

Univerzita Karlova v Praze

Filozofická fakulta

Ústav informačních studií a knihovnictví

Studijní program: Informační studia a knihovnictví

Studijní obor: Informační studia a knihovnictví

Bakalářská práce

Ondřej Smíšek

Značkovací jazyky a jejich použití

Mark-up languages and their usage

Praha 2011

Vedoucí práce: PhDr. Jan P o k o r n ý , Ph.D.

Oponent bakalářské práce:

Datum obhajoby:

Hodnocení:

Poděkování:

Na tomto místě děkuji za pomoc při tvorbě svému vedoucímu bakalářské práce PhDr. Janu Pokornému, Ph.D.

Prohlášení:

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně, že jsem řádně citoval všechny použité prameny a literaturu a že práce nebyla využita v rámci jiného vysokoškolského studia či k získání jiného nebo stejného titulu.

V Praze, dne 12. července 2011

.....
podpis studenta

Identifikační záznam:

SMÍŠEK, Ondřej. *Značkovací jazyky a jejich použití = Mark-up languages and their usage*. Praha, 2011. 64 s. Bakalářská práce (Bc.). Univerzita Karlova v Praze, Filozofická fakulta, Ústav informačních studií a knihovnictví. Vedoucí práce Jan Pokorný.

Abstrakt:

Tématem předkládané bakalářské práce jsou značkovací jazyky a jejich použití pro účinný přenos a ukládání informací. Značkovací jazyky se zrodily po dostatečném rozšíření výpočetní techniky mezi vědeckou obec. Jejich počátky lze datovat do první pol. 60. let 20. stol. První značkovací systémy vznikaly na univerzitách a vědeckých pracovištích v USA pro potřeby počítačové přípravy tiskových dokumentů. Později se objevilo jejich univerzální použití k popisu sémantického významu dokumentů i k popisu jejich formátových výstupů. S rozvojem internetu i kvalitní tiskové produkce připravované na počítačích rostl význam značkovacích systémů. Po vzniku webu lze mluvit doslova o boomu. V současnosti proniká značkovací jazyk XML do všech oblastí počítačového průmyslu pro svou univerzálnost, otevřenost a přizpůsobitelnost. Budoucnost webu podle všeho náleží multimediálnímu formátu HTML5.

Klíčová slova:

Značkovací jazyky, DTP, historie internetu, hypertext, WWW, syntaxe, sémantika, síťové protokoly, metadata, uživatelská rozhraní, kódování, multimédia.

Abstract:

The aim of this bachelor's thesis is mark-up languages and their usage in an effective transport and storage of information. Mark-up languages were created after expansion of a computer technology among scientists. The origins of languages can be dated at the first half of 60s in the 20th century. First mark-up systems were created at universities and science workplaces in the USA for desktop publishing needs. They were used for a semantic description and format output of documents after that. A development of the Internet and DTP meant growth of these languages. There was a boom when the Web was born. At present the XML language penetrates to all spheres of computer industry for its universality, openness and adaptability. The future of the Web probably belongs to the HTML5 multimedia format.

Keywords:

Mark-up languages, DTP, internet history, hypertext, WWW, syntax, semantics, network protocols, meta-data, user interfaces, coding, multimedia.

Obsah

PŘEDMLUVA.....	8
1 ÚVOD.....	11
1.1 ZAŘAZENÍ ZNAČKOVACÍCH JAZYKŮ.....	11
1.2 CESTA KE ZNAČKOVACÍM JAZYKŮM.....	11
1.3 CHRONOLOGICKÝ PŘEHLED VÝVOJE.....	15
2 ZNAČKOVACÍ JAZYKY 60. LÉTA – 1990.....	20
2.1 TYPSET, RUNOFF.....	20
2.2 GENCODE.....	21
2.3 GML.....	21
2.4 SCRIBE.....	22
2.5 TeX.....	23
2.6 DTD.....	26
2.7 PROPRIETÁRNÍ FORMÁTY.....	27
PostScript.....	28
SGML.....	29
RTF.....	31
3 ZNAČKOVACÍ JAZYKY 1991 – 2011.....	32
3.1 HTML.....	32
3.2 DocBook.....	36
3.3 VRML.....	37
3.4 XML.....	39
3.5 WML.....	40
3.6 SVG.....	41
3.7 SMIL.....	42
3.8 MATHML.....	43
3.9 RSS.....	44
3.10 XHTML.....	46
3.11 ATOM.....	47
3.12 FBML.....	48
3.13 HTML5.....	49
4 ZÁVĚR.....	55
SLOVNÍK ZKRATEK.....	57
BIBLIOGRAFIE.....	62
TIŠTĚNÉ PUBLIKACE V ČEŠTINĚ.....	62
TIŠTĚNÉ PUBLIKACE V ANGLIČTINĚ.....	63
ELEKTRONICKÉ ZDROJE V ČEŠTINĚ.....	64
ELEKTRONICKÉ ZDROJE V ANGLIČTINĚ.....	65

Předmluva

Cílem této práce je postihnout kontinuitu a směry vývoje značkovacích jazyků, které formují prostředí internetu. Z historického hlediska budou taktéž zmíněny jazyky používané v jiných počítačových oblastech (zejména v typografii), aby se zdůraznil jejich přesah.

Práce má následující strukturu:

1. Použití a potřeba značkovacích jazyků, jejich zařazení. Představení schopností popisu struktury a formátu Představení souvisejících technologií – hypertextu, internetu a jeho převládající služby *WWW*. Stručný vývoj širší počítačové oblasti v bodech.
2. Popis vybraných raných jazyků, jako předpoklad pro pochopení současného stavu. Jedná se o období do roku 1990, což je vymezení nejen formální pro potřeby této práce, ale slouží také jako připomenutí důležitého milníku mezi obdobím před webem a obdobím dominance webu.
3. Nejvýznamnější jazyky v období od roku 1991 do současnosti (červenec 2011). Představení konkrétních aplikací a možností.
4. Závěr. Pohled do nejbližší budoucnosti a zhodnocení tohoto směřování.

Tato bakalářská práce má 62 číslovaných stran textu a 23 obrázků. Citace v textu jsou zapisovány v hranatých závorkách pomocí prvního údaje záznamu (kapitálkami příjmení autora nebo jméno korporace), data vydání a čísla strany (Harvardský styl). Na konci práce lze nalézt logicky strukturovaný seznam použitých zdrojů, řazený dle abecedy a číslovaný. Citace jsou tvořeny dle platných norem *ČSN ISO 690* a *ČSN ISO 690-2*.

Typografické konvence jsou následující:

„Takto je zobrazena přímá citace.“

Takto je zobrazena citace v textu [AUTORITA, rok_vydání, strana].

Neproporcionálním písmem a s odsazením je zobrazen programový kód.

Důvodem zvolení tématu této bakalářské práce je můj dlouhodobý zájem o počítačové technologie, ke kterému studium na Ústavu informačních studií a knihovnictví (ÚISK) přidalo také zájem o informační vědu.

Téma značkovacích jazyků se vysloveně nabídlo. Překvapilo mě, že dosud (v posledních letech) na *ÚISK* nebylo zpracováno. Diplomanti se zabývali problematikou sémantického webu, metadatovými formáty, nebo on-line katalogy, ale ne značkovacími jazyky, tedy oblastí, jež je dnes pro internetové informační a knihovnické služby nepostradatelná.

Jasně volbě tématu napomohl také fakt, že jsem se během studia začal intenzívně věnovat návrhům a realizacím webových stránek. Rozvíjel jsem své kodéřské, designéřské a programátorské schopnosti. Přeci jen ale internet a web slouží především lidem, takže jsem rád, že mi *ÚISK* rozšířil obzory také v informační a humanitní oblasti, nehledě na skutečnost, že knihovnictví (alespoň ve světě) vždycky bylo jedním z nejdynamičtějších oborů integrujících počítačové technologie.

Nyní několik upozornění k samotnému textu. V práci se často střídá minulý čas s přítomným. Je to dáno popisným charakterem textu. Platí, že pokud jsou zmiňovány skutečnosti, které již nezasahují do současnosti, použit je minulý čas. Fakta zasahující do současnosti jsou zmiňována v přítomném čase.

V práci se vyskytují desítky počítačových zkratk. Zdálo se mi nepřehledné, abych vždy první výskyt zkratky vysvětloval v poznámce pod čarou. Proto byl seznam zkratk zařazen samostatně na konec práce.

Závěrem bych rád zmínil skutečnost, že tato bakalářská práce není a z principu nemůže být referenční příručkou popisovaných jazyků, tudíž nemůže ukázat všechny jejich možnosti, ale jen naznačit to důležité a popsat malé množství příkladů.

„Světová encyklopedie nemusí již v budoucnosti podle moderních představ vypadat jako řada dílů vytištěných a vydaných jednou pro vždy, ale jako určitý druh burzy myšlenek, místo, kde by se znalosti a nápady shromažďovaly, třídily, ujasňovaly, spojovaly a porovnávaly... Takováto encyklopedická organizace by nemusela být na jednom místě, ale mohla by mít formu sítě.“

Herbert George Wells¹

¹ WELLS, H. G. *World Brain*. 1st ed. London : Meuthuen & Co., 1938. 194 s.

1 Úvod

1.1 Zařazení značkovacích jazyků

Skutečnost, že symbol či krátká skupina znaků, obecně značka (*markup*, *tag*²), může představovat komplexní instrukci, je velmi působivá. Soubor takových značek, mezi kterými existují dohodnuté vztahy, tvoří umělý jazyk (*artificial language*). Podskupinou umělých jazyků, tedy takových, které vznikají vědomou dohodou lidí pro konkrétní účel, jsou informační jazyky (*information language*). Mezi ně řadíme jazyky programovací (*programming language*), selekční (*retrieval language*), dotazovací (*query language*), a také značkovací (*mark-up language*), jejichž významnými zástupci se chce zabývat tato práce.³

Značkovací jazyky působí a rozvíjejí se především v prostředí internetu. S jeho rozvojem a nutností dokonalejší struktury dokumentů, které obsahuje, vznikají také jejich nové verze, nebo se alespoň rozvíjejí ty stávající. Značkovací jazyky ovšem nejsou součástí pouze internetové služby *World Wide Web*, jak by se mohlo na první pohled zdát. To je nekritický pohled současníka. Značkovací jazyky ve skutečnosti vznikly o mnoho let dříve jako základ důležité oblasti počítačové typografie (*DTP*), což bude v této práci rovněž zmíněno. Raná počítačová typografie jednoduše rozvíjela funkce korektorů v tradičních tiskárnách [HAVLÍČEK, 2008, s. 8].

Značkovací jazyky lze dělit na popisné (*deskriptivní*) a výkonné (*procedurální*). Popisné jazyky jsou „pasivní“, jen předávají instrukce zpracovávajícímu programu, který určí, jak je vyhodnotit či zobrazit. Příkladem takového jazyka budiž např. *HTML*. Výkonné jazyky se svými vlastnostmi blíží spíše těm programovacím, lze v nich přesně řídit požadované výstupy, používat matematické operace, proměnné či správu paměti. Příkladem výkonného značkovacího jazyka může být např. systém *LaTeX*.

1.2 Cesta ke značkovacím jazykům

Příkazy strukturovaného dokumentu říkají, *co* se má udělat, a ne *jak* se to má dělat. Tím je zajištěna nezávislost na platformě, operačním systému, uživatelském rozhraní či grafickém systému [HLAVENKA aj., 2002, s. 7]. Na této jednoduché, ale silné myšlence jsou postaveny značkovací jazyky. Kombinují text a informace o něm, zápisem značek vysvětlují význam jeho jednotlivých částí a mohou (ale nemusí) určovat jejich vzhled [HAVLÍČEK, 2008, s. 7 – 9].

```
<h1>Tohle je opravdu zajímavý nadpis</h1>
```

Předchozí odstavec je ukázkou zápisu ve značkovacím jazyce. Tagy, které se nacházejí na začátku a na konci textu, říkají, že je to nadpis první úrovně. V tomto případě neříkají nic o tom, jak má vypadat. Vzhled dokumentu tak může být zcela oddělen od jeho významu, což je důležité pro strojové zpracování a dlouhodobou použitelnost dokumentu díky zpětné kompatibilitě.

Princip značkovacích jazyků se objevil v 60. letech. Na světě byl už první textový editor (viz kap. 2.1). Pro profesionální tisk dokumentů se používalo osvitových jednotek, které

² *Tag* z anglického visačka, jmenovka, štítek.

³ Zmíněné anglické ekvivalenty jsou dohodnutými termíny. Je možné je dohledat v Terminologické databázi knihovnictví a informační vědy (*TDKIV*) – kterou spravuje Národní knihovna – na následující adrese: http://www.nkp.cz/o_knihovnach/Slovník/index.htm

dostaly spolu s dokumentem sadu příkazů, jak tento obsah zpracovat. Dokument osvitová zařízení přenesly na film, ze kterého ho již mohly zpracovat tiskárny. Tehdejší roztržitost technologií, množství výrobců a živelný vývoj znemožňovaly vzájemnou kompatibilitu tiskových procedur. Ve druhé polovině 60. let však W. Tunnicliffe pracoval na systému *Gen-Code*, který by mohl oddělovat obsah dokumentu od jeho formy.

Z teoretických poznatků tohoto výzkumníka vyšli Ch. F. Goldfarb, E. Mosher a R. Lorie z *IBM*, kteří správně uchopili problém a v letech 1969 – 1973 stvořili první obecný a univerzálně použitelný značkovací jazyk – *GML*. Poté se vývoje ujaly standardizační organizace a další sdružení, jejichž společným výsledkem byl jazyk *SGML*, který se stal mezinárodní normou.

V době raného internetu, který obsluhovaly skupiny armádních inženýrů⁴ a vědeckých pracovníků, nešlo vůbec mluvit o nějaké možnosti strukturování informací. Jednalo se o přelom 60. a 70. let, a už samotný fakt, že fungovala elektronická zpravodajská síť, která byla nezávislá na připojených uzlech a rychlost dat v ní putujících činila až tisíce znaků za vteřinu, působil dostatečně odvážně. Nebyla žádná jednotná platforma zobrazování, protože většina výrobců počítačů měla vlastní uzavřené systémy. Výstupy na obrazovku doplňovaly tiskárny a děroštitkové stroje. Internet obsahoval textové zprávy, které se posílaly pomocí jednoduchých protokolů (zastřešených transportní vrstvou *NCP*) mezi počítači [viz např. ČESENK, 1995, s. 95 – 98].

Od konce 70. let se obrazovka stala dominantním výstupem i pro internetové aplikace. Vznikající veřejné služby představovaly terminálové výstupy z různých databází.⁵ Výrazným milníkem ve vývoji internetu byl přechod z protokolu *NCP* na protokoly *TCP/IP* v roce 1983. Stávající *NCP* již nemohl poskytovat rozvíjející se síti potřebnou základnu [NAUMANN, 2009, s. 355]. Zejména populární média tento okamžik považují za vznik „skutečného“ internetu.

Velmi oblíbené se staly aplikace *BBS*, které v jednoduchém rozhraní zastřešovaly protokoly elektronické pošty, sdílení souborů, elektronických nástěnek (konferencí), chatovacích služeb či síťových her [ČESENK, 1995, s. 63 – 70]. Obrazovky terminálů a začínajících *PC* měly textový výstup, který zobrazoval v monochromatickém režimu jediný typ písma napevno definovaný v paměti. Toto písmo bylo monotypické, takže každý jeho znak zaujímal na obrazovce stejně velkou pozici, aby zapadl do obrazové matrice.⁶ Do poloviny 80. let si na internetu uživatelé museli vystačit s tímto schématem. Pořád to byl především věk podnikových sálových (*mainframe*) a skříňových počítačů (*mini-computer*).

Firma *Apple* uvedla svůj osobní počítač na trh v roce 1976, *IBM* roku 1981. Obě platformy byly zpočátku příliš neohrabané a slabé na to, aby se mohly zapojit do rozvoje internetu. Počet jeho uživatelů však utěšeně rostl, ze stovek se stávaly tisíce⁷ a mohly se už projevat zájmy některých vědeckých skupin o to, jak chtějí strukturovat své informace.

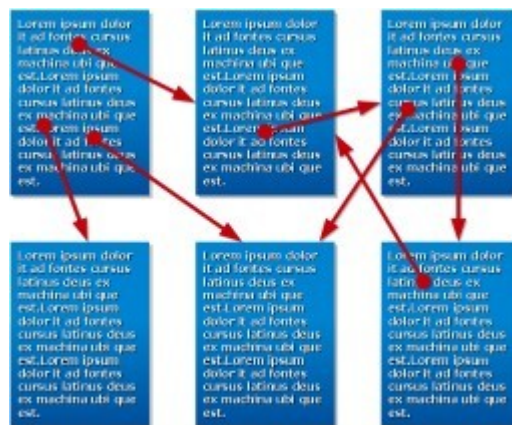
Mezitím se rozvíjely další oblasti výpočetní techniky a teorie komunikace, které budou za pár let základními součástmi nového projektu internetu – *World Wide Webu*. Koncem 80. let již dostatečně výkonné *PC* stanice zaplavily všechny kanceláře a *GUI* nebylo pouhým snem, takže jednoduché textové výstupy začaly působit značně omezeně.

⁴ Projekt *ARPANET* byl dobře popsán např. zde: <http://www.earchiv.cz/a95/a504c502.php3>

⁵ Např. bankovní systémy, rezervace letenek a bibliografické soupisy.

⁶ Typickou konfigurací bylo na 24 řádcích 40 znaků alfanumerických znaků (mřížka 24x40). Pozdější *PC* od *IBM* mělo mřížku 25x80.

⁷ Vyčerpávající tabulka počtu připojených uzlů zde: <http://www.earchiv.cz/a95/a504c50b.php3>



Obr. 1: Schéma hypertextové struktury mezi šesti dokumenty – uzly (*nodes*) se sedmi odkazy (*links*) [získáno: 03-18-2011; zdroj: autor].

Princip hypertextu,^{8,9} aniž by to sám tušil, popsal už H. G. Wells.¹⁰ Po 2. sv. válce myšlenku dále prakticky rozvinuli V. Bush (projekt *Memex*)¹¹ [NIELSEN, J., 1990, s. 29 – 32], D. Engelbart¹², T. Nelson (projekt *Xanadu*)¹³ [NIELSEN, J., 1990, s. 33 – 35; VLASÁK, 1993, s. 149 – 153] a další.¹⁴ Oblíbené *BBS* uměly v té době běžně zobrazovat obrázky ve formátu *GIF*.

Jako vrchol hypertextové technologie před příchodem webu budiž představen program *HyperCard*. V roce 1987 ho uvedla na trh firma *Apple*. Nabízela jej zdarma ke každému no-

⁸ Hypertext na rozdíl od „obyčejného“ textu nemá žádné předem určené pořadí zobrazování informací [viz např. [45], s. 1]. Hypertext je nějaký informační systém, který zobrazuje informace zároveň s odkazy na jejich upřesnění, rozšíření nebo zdroje. Pořadí hypertextových dokumentů utváří sám čtenář tím, jak informace z nich vybírá. Skládá tak vždy nový dokument dle svého zájmu (obr. 1). Výsledkem je provázání obrovského množství znalostí a možnost volně se mezi nimi pohybovat (princip funkce „historie“ internetových prohlížečů).

⁹ Na *ÚISK* psal v r. 2006 o hypertextu např. J. Müller diplomovou práci s názvem „Hypertext a hypermédia ve vyhledávacím procesu“. Více informací zde: <http://isd.p.alstanet.cz/PublicationView.aspx?PublicationID=267>

¹⁰ Viz motto této kapitoly.

¹¹ Návrh elektromechanického znalostního zařízení, pracujícího na principu hypertextu. Bush jej publikoval v roce 1945 jako článek „As We May Think“ v časopise *The Atlantic*. Plný text je volně dostupný na stránkách časopisu: <http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/3881/>

¹² Velmi plodný odborník. Mimo jiné autor teorie hyperlinků, počítačového zpracování textu, nebo také vynálezce polohového zařízení – myši.

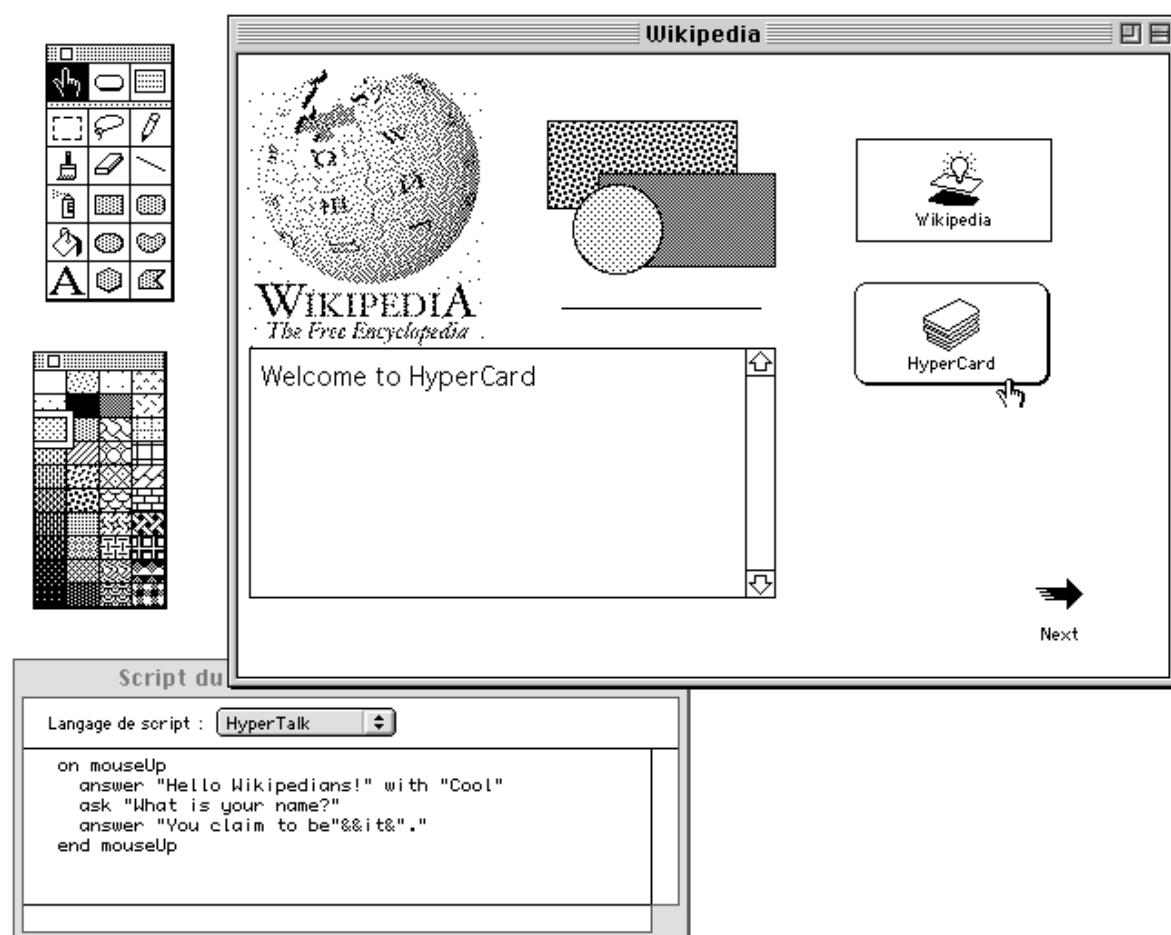
¹³ Hypertextový systém, který měl obsahovat úplné lidské vědění. Nebyl nikdy dokončen, nicméně inspiroval mnoho dalších odborníků; jeho vizi poněkud zjednodušil a převzal Berners-Leeův *WWW*. Že je až příliš zjednodušená, Nelson popsal odměřenými slovy: „*HTML je přesně to, čeho jsme se chtěli uchránit – věčné rozbité odkazy, odkazy, které jsou pouze ven, citáty, jejichž původ nemůžete najít, žádný management verzí, žádný management práv...*“ Ukázka hypertextového systému je dostupná ke stažení pro platformu *Windows* zde: <http://xanarama.net/>

¹⁴ Je zajímavé sledovat, že po uvedení Bushova článku se téměř dvacet let na poli hypertextu nic nedělo, zato jen v 60. letech vzniklo hned několik pokrokových implementací. Kromě zmíněného *Xanadu* ještě *oN-Line System* (NLS), *Hypertext Editing System* (HES, 1967) a *File Retrieval and Editing System* (FRESS, 1969). V 70. letech stojí za zmínku nekomerční počítač *Xerox Alto* (1973), jenž jako první používal Engelbartovu myš a *GUI*. Tímto strojem postaveným v laboratoři *Xerox PARC* se později inspiroval Steve Jobs z *Apple* při návrhu počítače *Macintosh*. Dále vznikl *PROMIS* (*Problem-Oriented Medical Information System*) ovládaný pomocí dotykové obrazovky nebo *Aspen Movie Map* (laboratoř MIT), systém virtuální procházky po městě Aspen v Coloradu. Stojí dále za připomenutí, že za většinou projektů tzv. *MIT Media Lab* se skrývá jméno vynikajícího teoretika i praktika nových médií a počítačových technologií N. Negroponteho (více viz literatura [22]).

vému počítači *Macintosh*. Kombinace nulové ceny a vynikající funkčnosti stála za jeho vysokou popularitou [NIELSEN, 1990, s. 93 – 97].

HyperCard si moderní čtenář může představit jako soubor nástrojů pro tvorbu komplexních dokumentů ve stylu programů *Microsoft PowerPoint* či *OpenOffice Draw*. Je možné v něm tvořit libovolně strukturované texty s obrázky a vše následně provázat hypertextovými odkazy. Aplikace v *HyperCard* může být rozsáhlá encyklopedie, stejně dobře jako osobní organizér. Vše je navíc velmi zábavné a jednoduché díky integrovanému programovacímu jazyku *HyperTalk*.

V 80. letech zažíval hypertext zřejmě své vůbec neplodnější období.¹⁵



Obr. 2: Ukázka rozhraní aplikace *HyperCard*

[získáno: 02-04-2011; zdroj: Příspěvatelé Wikipedie. Dostupné z WWW: <http://cs.wikipedia.org/>].

V roce 1989 přišel na svět *World Wide Web*, aplikace běžící přes protokol *HTTP*, která je dnes nejrozšířenější službou na internetu a mnoho lidí se mylně domnívá, že je vlastně internetem samotným. Je to informační systém, který v sobě spojuje text, grafiku, hypertextové odkazy a další prvky [PÍSEK, 2006, s. 11]. Pomocí odkazů se uživatel může libovolně „procházet“ těmito multimediálními reprezentacemi dokumentů uložených na serverech po celém světě. Po každém zadání *URL* protokol *HTTP* pošle požadovanou *HTML* stránku či *XML* aplikaci uživateli a klientský prohlížeč (*browser*) jí po zpracování zobrazí.

¹⁵ Opět telegraficky: EDS (*Electronic Document System*) – hypertextový systém na Brownově univerzitě pro interaktivní technické manuály. 1980 – Wes Kussmaul vytvořil první internetovou encyklopedii. 1981 – počítač *Xerox Star* spojuje GUI, myš, Ethernet. 1984 – projekt *NoteCards* (opět *Xerox PARC*) představuje virtuální hypertextovou kartotéku. 1985 – aplikace *Intermedia* je hypertextová databáze, nástupce projektů HES a FRESS.

Výše popsany fenomén navrhl a uvedl do provozu T. Berners-Lee se svými kolegy z laboratoří *CERN*.¹⁶ Bylo to spojení již existujících technologií, principů a dříve popsanych nápadů zmíněných v předminulém odstavci této práce. Někteří lidé si proto nemyslí, že by Berners-Lee, obecně vyzdvihovalý za svou geniální myšlenku, nesl oprávněně takový díl zásluh.¹⁷

Vědci z *CERN* sledovali jednoduchý cíl: chtěli výsledky své práce rychleji, jednodušeji a efektivněji sdílet s lidmi z dalších výzkumných ústavů, které byly také připojeny k internetu. Do té doby se používala elektronická pošta (protokoly *POP* a *SMTP*), *Telnet*, *Usenet* (*Netnews*), či jednoduchý přenos souborů mezi servery (protokol *FTP*) [VLASÁK, 1999, s. 40; NAUMANN, 2009, s. 354; ČESENĚK, 1995, s. 96 – 98]. Tyto funkce byly pochopitelně výborné a používají se dodnes,¹⁸ nejsou však tolik intuitivní a bezprostřední. Pro popis dokumentů sloužily různé verze zbytečně komplikovaného jazyka *SGML*, dále se používal *TeX*¹⁹ nebo *PostScript*.

Berners-Lee využil protokol *HTTP*, na němž vystavěl aplikaci pro přenos hypertextových souborů napsaných ve značkovacím jazyce, který vycházel ze staršího *SGML*, aby posléze dostal vlastní označení – *HTML*. Vytvořil i první prohlížeč a editor těchto dokumentů v jednom, který pojmenoval *WorldWideWeb*, později nazývaný *Nexus*, aby se nepletl přímo s aplikací *WWW*.

Jak se brzy ukázalo, byla to trefa do černého.²⁰ Univerzalita *HTML* stránek ukládaných v běžném *ASCII* textovém souboru znamenala nevyhraněnost na platformě [DELLWIG, 2000, str. 28]. Poskytnutí prohlížeče a *WYSIWYG* editoru stránek k volným úpravám (*free-ware*) iniciovalo okamžité vytvoření nadšené komunity, která začala vyvíjet vlastní produkty. *Nexus* byl totiž poměrně krkolomný na ovládání, navíc ještě nezobrazoval grafiku (obr. 3).

1.3 Chronologický přehled vývoje

V této poslední části úvodní kapitoly následuje v bodech představený vývoj značkovacích jazyků, nejdůležitějších technologií souvisejících s *DTP*, internetem a *WWW*, dále produktů (zejm. prohlížečů), jakožto i významných událostí.

- **1945:** V. Bush navrhuje *Memex*.
- **50. léta:** Jako alternativní výstupní zařízení pro *mainframe* se kromě stávajících děroštítkových, děrnopáskových a magnetopáskových jednotek či tiskáren, zavádí monitor a příkazové ovládací rozhraní (*CLI*).
- **1960:** Počátek projektu *Xanadu*.
- **1963:** Vydán první *ASCII* standard pro kódování znaků anglické abecedy.

¹⁶ Možná stojí za to připomenout, že je to jedna z mála internetových technologií, která nevznikla na území Spojených států.

¹⁷ Velmi důkladná kritika *W3C* v podání českého webdesignéra a programátora Chamurappiho (své skutečné jméno nikde neuvádí) je k dispozici zde: <http://www.webylon.info/>

¹⁸ I když se mnohé změnilo. Například lehce napadnutelný protokol *Telnet* je nahrazen šifrovaným *SSH*.

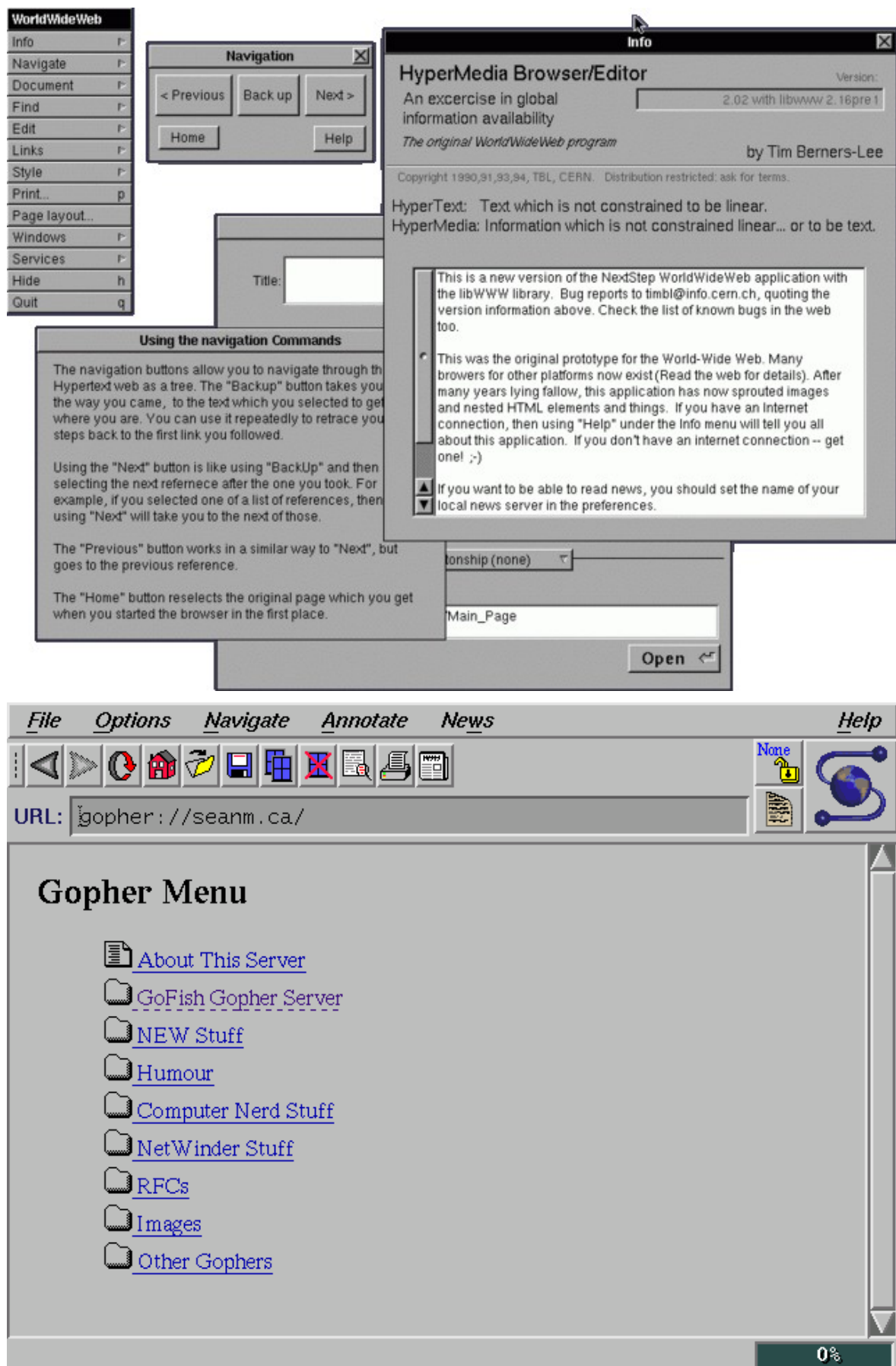
¹⁹ Název pochází z řeckého *τεχνη*, dovednost, čte se tedy [tech], nikoli [teks].

²⁰ Nabízí se otázka, proč služba *WWW* byla tak úspěšná, zatímco *Gopher*, jeho přímý (taktéž hypertextový) konkurent vytvořený v roce 1991 na Minnesotské univerzitě, po několika letech upadl do zapomnění. Pravděpodobně se tak stalo proto, že *Gopher* byl sice v dobré víře vytvořen akademiky, ale již v době, kdy na internetu začali převládat „běžní konzumenti z davu“, pro které byl systém s téměř nulovými grafickými možnostmi neatraktivní. V roce 1993 navíc *Gopher* podlehl zpoplatnění. Dva roky poté již webové služby internetu zcela dominovaly. Dnes *Gopher* servery v podstatě neexistují a pokud ano, jsou to spíše drobné projekty osamělých nadšenců.

- **1964:** J. H. Saltzer představuje první počítačový textový editor *TYPSET*.
- Zahájení projektu *Multics*.
- **1965:** T. Nelson zavádí termín hypertext.
- **1967:** W. Tunnicliffe zahajuje projekt *GenCode*.
- **1969:** Spuštěn *ARPANET*.
- Z ukončeného *Multicsu* se rodí *Unix*.
- *Xerox* představuje kopírku.
- **1970:** Komerčně dostupná jehličková tiskárna.
- **1973:** V laboratořích *IBM* je dokončen značkovací jazyk *GML*.
- *Xerox* představuje koncept *GUI* nazvaný *WIMP*.
- **1975:** Komerčně dostupná laserová tiskárna.
- **1976:** *Apple* představuje použitelný osobní počítač.
- **1978:** D. E. Knuth uvádí typografický značkovací formát *TeX*.
- **1979:** Pro komerční užití je dostupný systém *Scribe* B. K. Reida.
- **1981:** *IBM* zahajuje prodej *PC*.
- **1983:** Internet přechází z *NCP* na síťový protokol *IP* a transportní protokol *TCP*.
- **1984:** První laserová tiskárna pro domácí použití od *HP*.
- *Apple* zavádí *OS* založený na *GUI*.
- **1985:** *Adobe Systems* uvádí na trh typografický jazyk *PostScript*, jenž nalézá nečekané užití také pro tvorbu internetových souborů.
- *Microsoft* představuje *OS WINDOWS 1.0*.
- **1986:** Obecný značkovací jazyk *SGML* se po dlouholetém vývoji stává normou *ISO 8879:1986*. *SGML* umožňuje definici vlastních podřízených jazyků na základě *DTD*.
- **1987:** *Microsoft* představuje univerzální textový formát *RTF*.
- **1989:** Zahájen projekt *World Wide Web*.
- **1990:** *ARPANET* je pohlcen ostatními sítěmi a zaniká.
- **1991:** Zveřejnění prvního textového prohlížeče webu, a také editoru stránek.
- Jako konkurence webu vzniká systém *Gopher*.
- **1993:** Američtí studenti M. Andreessen a E. B. B. uvádějí prohlížeč *Mosaic*. Je volně dostupný, podporuje první verzi *HTML*, zobrazuje text i grafiku v jednom okně a ovládá se pouze myší. Od této chvíle lze mluvit o skutečném boomu internetu.
- Jdoucí proti proudu se může zdát zrod textového prohlížeče *Lynx*, který je nicméně úspěšně vyvíjen doposud a představuje např. pro vývojáře neocenitelnou pomůcku.
- Na trh taktéž přichází nástupce *PostScriptu* – *PDF*.
- **1994:** Berners-Lee zakládá neziskovou organizaci *W3C*. Potřeba řízených pravidel pro popis internetových dokumentů je zcela legitimní.
- Autoři *Mosaicu* zakládají vlastní společnost *Netscape* a vydávají prohlížeč *Navigator*.
- Norští studenti, nespokojení s chováním *Nexusu*, vyvíjejí vlastní prohlížeč *Opera*, který existuje doposud.
- **1995:** Organizace *IETF* vydává specifikaci *HTML 2.0* – webové formuláře.

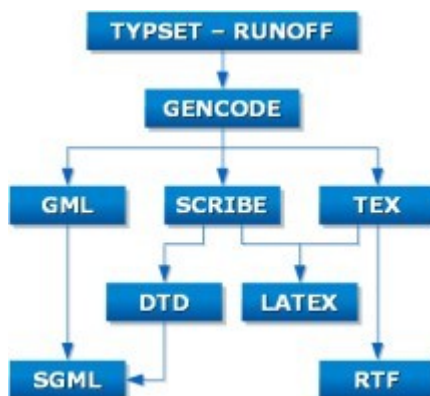
- Netscape přidává tabulky a rámy (zobrazení více *HTML* dokumentů v jednom okně). Těto verzi se říká *HTML 3.0*, i když oficiálně neexistuje.
- Microsoft, dosud spící na poli internetu, vydává prohlížeč *Internet Explorer*, jehož jádro přebírá z *Mosaicu*.
- Vychází *JavaScript*, objektový klientský jazyk pro *WWW*.
- **1996:** Pro definování vzhledu dokumentu je zaveden jazyk *CSS (Level 1)*. Nejlépe ho interpretuje *Internet Explorer*.
- Microsoft (spíše neúspěšně) prosazuje vlastní klientské skriptovací jazyky *JScript* a *VBScript*.
- **1997:** Vychází specifikace *HTML 3.2* jako první významný počín *W3C*.
- Microsoft představuje *DHTML* – spojení *HTML*, *DOM*, *CSS* a skriptovacích jazyků.
- *Internet Explorer* dosahuje velké popularity, i když z části tím, že je předem instalován do operačních systémů *Windows* i *MacOS*.
- **1998:** *Netscape Navigator* je zcela poražen *Internet Explorerem*. Netscape uvolňuje zdrojový kód svého prohlížeče. Vzniká projekt *Mozilla Suite*, který se ale potýká s řadou problémů.
- Představuje se revoluční *HTML 4.0*. Nabízí rozšířené možnosti tabulek a formulářů, vkládání objektů, skriptů a stylů, podporu národních zvyklostí a další.
- Uvedeno *XML 1.0*.
- Uvedeno *CSS Level 2*.
- **1999:** Vychází revize *HTML 4.01*, která se používá běžně dosud. Budoucí vývoj se má podle stanoviska *W3C* týkat pouze *XHTML* a *XML*.
- Objevuje se mnoho jazyků založených na *XML*.
- **2000:** Vychází specifikace *XHTML 1.0*.
- *Internet Explorer* je nainstalován na více než 90% *PC* připojených k internetu, a to i přesto, že obsahuje bezpečnostní mezery a různě si vykládá zavedená pravidla.
- **2001:** Vychází revize *XHTML 1.1*.
- **2002:** Využití *XHTML* sice stoupá, ale je zřejmé, že budoucnost mu nepatří. *W3C* nedokáže adekvátně reagovat na potřeby vývojářů.
- **2003:** Apple představuje vlastní prohlížeč *Safari*. Ukončuje tím spolupráci s *Microsoftem*, který pro Apple dodával *Internet Explorer*.
- **2004:** Projekt *Mozilla Suite* je pro neúspěch ukončen, ujímá se ho skupina s názvem *SeaMonkey*. Původní prohlížeč slouží k vývoji nového *Mozilla Firefoxu*. Ten nabízí rychlé a jednoduché rozhraní, záložky, správce stahování, pokročilý prohlížeč kódu a možnost přidávání doplňků. Rychle si získává popularitu.
- Za přispění firem *Google*, *Mozilla*, *Opera* a *Apple* vzniká sdružení *WHATWG*, aby řešilo akutní potřebu vylepšení *HTML/XHTML* platformy.
- Vychází revize *XML 1.1*.
- **2005:** Představuje se soubor technologií *AJAX* pro vývoj interaktivních webových aplikací.
- Počátek vývoje *CSS Level 3*. Specifikace je rozdělena do tzv. modulů, aby mohly být hotové části vydávány postupně.
- **2006:** Po pěti letech „vývoje“ vydává Microsoft nový *Internet Explorer*. Je to jedna z nejhorších verzí vůbec.
- **2007:** Na základě aktivit *WHATWG* je *W3C* založena pracovní skupina pro specifikaci *HTML5*.

- **2008:** *Google* představuje rychlý a uživatelsky jednoduchý prohlížeč *Chrome*. Okamžitě se řadí mezi pěti nejvíce používaných.
- **2009:** Zrušen vývoj *XHTML 2.0*.
- **2010:** Vývojáři začínají aplikovat prvky z *HTML5*. *Microsoft* ohlašuje, že nový *Internet Explorer* bude podporovat moderní standardy.
- **2011:** Počátkem roku se představují nové verze prohlížečů *Internet Explorer*, *Chrome* a *Firefox*. Vývoj ostatních prohlížečů směrem k podpoře moderních technologií je také urychlen. Aplikace *HTML5* a *CSS 3* je na dobré úrovni.



Obr. 3: Internet na poč. 90. let. V horní části *WorldWideWeb (Nexus)* zobrazující některou z prvních webových stránek [získáno: 01-20-2011; zdroj: Příspěvatelé Wikipedie. Dostupné z *WWW*: <http://cs.wikipedia.org/>]. Ve spodní části ukázka *Gopher* serveru v grafickém prohlížeči *NCSA Mosaic* [získáno: 01-20-2011; zdroj: autor].

2 Značkovací jazyky 60. léta – 1990



Obr. 4: Geneze nejdůležitějších značkovacích a typografických systémů do roku 1990
[získáno: 02-05-2011; zdroj: autor].

2.1 TYPSET, RUNOFF

V roce 1964 mladý počítačový odborník J. H. Saltzer naprogramoval jednoduchý textový editor *TYPSET* a stal se tak průkopníkem počítačové typografie. Nadstavbou *TYPSETu* byla ještě aplikace *RUNOFF* sloužící jako výstupní procesor. Není bez zajímavosti, že *RUN-OFF* posloužil pro další podobné programy na tehdy se rodící platformě *Unix*.²¹ Jinak pracoval na mnoha sálových (*IBM 360*, *IBM 370*) a skříňových počítačích (*DEC*), přičemž byl také součástí systému *Multics*.²²

RUNOFF nabízí základní formátovací procedury, jako stránkování, nadpisy, seznamy, odstavce, viz ukázka:²³

```
.SK 1
Bezny text, který následuje dvoupolozkový seznam:
.TB 4
.OF 4
.SK 1
1. #První položka
.OF 4
.SK 1
2. #Druhá položka
.OF 0
.SK 1
```

Po zpracování procesorem může výstup vypadat nějak takto:

```
Bezny text, který následuje dvoupolozkový seznam:
1. První položka
2. Druhá položka
```

²¹ Jedná se o textové editory *roff* a *nroff* a textový procesor *troff*. Ten poslední jmenovaný je v současnosti nahrazen procesorem *groff*, jenž je vyvíjen jako svobodný software.

²² *Multics* byl první OS umožňující sdílení výpočetního času pro více uživatelů najednou. Byl důležitým předpokladem pro vznik terminálových služeb, ačkoliv zůstal pro malý komerční zájem nedokončen. Jeho nástupcem se stala platforma *Unix*.

²³ Česká ukázka je bez diakritiky pro navození dobové situace, protože značkovací systémy 60. a 70. let vzhledem k jejich omezení na americký kontinent (příp. některé západoevropské země) nezaváděly více jak 7bitovou *ASCII* tabulku (128 znaků, bez znaků národních abeced).

2.2 GenCode

O tři roky později W. Tunnicliffe na setkání *GPO* představil myšlenku *generic coding*²⁴ pro dokumenty připravované k tisku elektronicky. Uvědomoval si, že je třeba vytvořit obecnou platformu pro oddělení obsahu od formy, a tím se vyhnout nekompatibilitě mezi různými zařízeními. Několik let se Tunnicliffe zabýval sémantickým systémem nazvaným *GenCode*, který posloužil za vzor dalším projektům.

2.3 GML

Z myšlenek W. Tunnicliffa vyšel Ch. Goldfarb, jenž roku 1966 nastoupil do *IBM* coby čerstvý absolvent práv. Příprava dokumentů k tisku se mu zdála neefektivní a zdlouhavá, protože musel řešit velké problémy s laděním jednotlivých systémů. Nejvíce ho přitom trápilo, že každá aplikace používala zvláštní zápisy a značky.

Od roku 1969 začal s kolegy E. Mosherem a R. Lorieem navrhovat obecný jazyk, který by situaci zjednodušil. Cesta to nebyla přímá, jazyk několikrát měnil i jméno, jak postupně rostl (např. *Integrated Text Processing*). Vedení *IBM* si naštěstí uvědomovalo, o jak důležitý projekt se jedná, a tak uvolňovalo další prostředky na vývoj. Z jednoduché potřeby jednoho člověka ulehčit si práci, během tří let vznikl produkt nesmírné důležitosti.

V roce 1973 byl jazyk pojmenovaný *GML* představen veřejně. Dostalo se mu kladného přijetí, zaujal státní správu i velké soukromé podniky. Goldfarb si vysloužil vysokou pozici a *GML* se dál rozrůstal až do roku 1978, kdy uzrál čas na *SGML*. Toho roku ještě vznikl důležitý syntaktický jazyk *DTD* pro popis značek, který našel užití právě při aplikaci *SGML*.

Proti svému pokračovateli je *GML* stále poměrně jednoduchý značkovací jazyk, jak může doložit následující ukázka:

```
:h1.Značkovací jazyky a jejich použití
:p.Napsal Ondrej Smisek
:p.Prace ma nasledujici cleneni:
:ol
:li.Uvod
:li.Stat
:li.Zaver
:eol.
:p.To by mohlo byt vse.
```

Tiskový výsledek bude zhruba následující:

ZNACKOVACI JAZYKY A JEJICH POUZITI

```
Napsal Ondrej Smisek
Prace ma nasledujici cleneni:
- Uvod
- Stat
- Zaver
To by mohlo byt vse.
```

²⁴ Všeobecně použitelného, zobecnělého zápisu.

2.4 Scribe

Projekt *Scribe* byl v době svého vzniku nejpokročilejším typografickým jazykem ve smyslech schopnosti oddělení obsahu dokumentu od jeho formy a doplňování automaticky generovaného textu. Několik let na něm pracoval počítačový odborník B. K. Reid.²⁵ V roce 1980 obhájil svůj projekt jako disertaci, o rok dříve už ale *Scribe* nabízel ke komerčnímu užití.

Scribe stál jako vzor mnoha budoucím značkovacím systémům, jeho přímý následovník je *LaTeX* (viz kap. 2.5). Největšími přínosy tohoto jazyka jsou myšlenky oddělení definice vzhledu od samotného značkování dokumentu a deklarace jednoznačných gramatických pravidel pro popisné prvky. Podobnost s externím stylpisem novějšího CSS není náhodná, ale pochází skutečně odsud. Z makropříkazů pro vytváření kapitol, stránek nebo poznámek pod čarou vyšel zase programovací jazyk *JavaScript*.

Tvorba dokumentu ve *Scribu* sestává z několika jednoduchých kroků, které budou současným tvůrcům internetových stránek velmi povědomé. Nejdříve se napíše vlastní text v libovolném editoru. Podle potřeby a funkce se následně do textu přidají popisné značky. Tím je soubor připraven ke kompilaci, po které získá svou finální podobu. Kompilátor používá styly definující vzhled všech logických prvků. Takových stylů lze mít libovolné množství. Kompilovaný soubor se zpracuje včetně všech makropříkazů vynucených typografických funkcí, a může jít např. k tisku.

Pro zjednodušení práce a větší přehlednost v delších textů kodér může každou kapitolu zapsat do samostatného souboru. Ty pak zastřeší jeden „meta-soubor“ s patřičnými vazbami, definicemi vzhledu a makry.

Tagy ve *Scribu* jsou uvozeny zavináčem (@). Samotné značky lze od výrazů oddělovat libovolnými typy závorek nebo uvozovek. Také jsou celkem tři možnosti zápisu značek: dlouhá, krátká a nejkratší forma. Dlouhá forma obsahuje počáteční a koncový tag, krátká pouze počáteční. Nejkratší zápis pomocí jediného tagu na jednom řádku lze s výhodou užít pro velmi krátké výrazy.

Nejprve ukázka dlouhé formy zápisu citace:

```
@Begin(Quotation)
  Cogito ergo sum.
@end(Quotation)
```

Nyní krátký zápis:

```
@Quotation(
  Cogito ergo sum.
)
```

A nejkratší zápis:

```
@Quotation(Cogito ergo sum.)
```

Ve všech třech případech bude výstup následující:

```
Cogito ergo sum.
```

Scribe nabízí značky pro zápis citací, veršů, počítačového kódu, centrovaného textu, oboustranně zarovnaného textu, číslovaných a nečíslovaných seznamů, matematických teorémů,

²⁵ Tvůrce sám *Scribe* popsál jako „*document specification language and its compiler*“.

rovníc,²⁶ glosářů aj. Umí pochopitelně zvýraznit písmo (tučné, kurzíva atd.). Ze sémantických značek lze připomenout různé hlavičky, kapitoly, patičky, tabulky a sloupce. Automatizace v podobě počítání stránek, názvů kapitol, poznámek pod čarou či rejstříků dosáhla ve své době velké propracovanosti a výborné použitelnosti.

Na závěr komplexnější ukázka kódu:

```
@Begin(UkazkovaStrana)
@MajorHeading(Znackovaci jazyky a jejich pouziti)
@CopyrightNotice(Ondrej Smisek)
@SubHeading(Osnova:)
@Begin(enumerate)
Uvod
Stat
Zaver
@End(enumerate)
@Begin(Display)
Tady je nejaky delsi text napsany ve Scribu@Foot(Jehoz tvurcem je B.
K. Reid.) na ukazku. Je zde take vyborne videt, ze se slova umi
zarovnat s pravym i levym okrajem. Ovsem Scribe toho umi mnohem vic!
Tato bakalarska prace ale neni referencni priruckou, takze timto
ukazka konci.
@End(Display)
@PageFooting(center[Strana @ref{page}])
@End(UkazkovaStrana)
```

A výsledek:

ZNACKOVACI JAZYKY A JEJICH POUZITI

Ondrej Smisek

Osnova:

1. Uvod
2. Stat
3. Zaver

Tady je nejaky delsi text napsany ve Scribu¹ na ukazku. Je zde take vyborne videt, ze se slova umi zarovnat s pravym i levym okrajem. Ovsem Scribe toho umi mnohem vic! Tato bakalarska prace ale neni referencni priruckou, takze timto ukazka konci.

- - - - -

¹ Jehoz tvurcem je B. K. Reid.

2.5 TeX

TeX je typografický program, který zpracovává napsaný text do vysázené podoby. Nabízí vynikající možnosti pro zápis matematických vzorců, méně běžných písem, speciálních znaků, tabulek či hudební notace, které neposkytují ani moderní *WYSIWYG* editory. Z dosud představených technologií je jasně nejvyspělejší. Nedílnou součástí *TeXu* je navíc program *METAFONT* – mocný vektorový editor, původně určený pro návrh sad písem, nicméně s oblibou užívaný také na tvorbu *TeXové* grafiky.

Ve vstupním (zdrojovém) souboru běžného *ASCII* formátu s příponou *.tex*, jsou kromě sázeného textu také značky dvou druhů. Ty, které popisují vzhled dokumentu jsou značky optické, zatímco značky logické popisují logickou strukturu (smysl) dokumentu. V *TeXu*

²⁶ I když v této oblasti je současník *Scribu TeX* kvalitativně úplně jinde.

se značkám říká řídicí příkazy, případně makropříkazy [BALVÍNOVÁ – BÍLÝ, 1994, str. 7]. Obsah spolu s typografickými značkami může zpracovávat procesor nazývaný *ConTeXt*.

TeX vytvořil americký matematik D. E. Knuth v roce 1978. Motivací pro takový krok mu byla skutečnost, že kvalita sazby jeho knih (zejména pak matematických vzorců) neodpovídala jeho představám. Kolem systému okamžitě vznikla nadšená komunita. Aby bylo ušlechtlé zpracování textu, byly vytvořeny různé soubory maker (*PlainTeX*, *LaTeX*, atd.). Na přípravu vědeckých publikací v té době neexistovala lepší sazební forma. Kromě toho se *LaTeX* uchytil také na internetu pro popis dokumentů.



Obr. 5: Loga *METAFontu*, *TeXu* a jeho maker, která demonstrují některé jejich typografické možnosti [získáno: 04-02-2011; zdroj: Wikipedia contributors; dostupné z WWW: <http://en.wikipedia.org/>].

TeX neomezuje žádná platforma, což je důležitý předpoklad pro univerzálně použitelný jazyk. Existuje velké množství volně dostupných a komerčních editorů pro platformy *Unix*, *Linux*, *Mac OS*, *DOS*, *OS/2* a *Windows*. Dokumenty jsou zpracovávány dávkově, což je výhodné z hlediska nízkých hardwarových nároků na systém [RACHŮNEK, 2003, str. 7, 9]. Vyžaduje to ovšem vyšší počítačovou gramotnost obsluhy, což je hlavní důvod toho, že *WYSIWYG* editory jsou dnes mnohem rozšířenější.

Do textu, který má být vysázen, se přidávají samotné *TeXové* příkazy, jež uvozuje zpětné lomítko (\) nebo jiný speciální znak. Složené závorky ({}) pak seskupují parametry maker. Nechybí ani možnost komentáře (%), tedy textu, který je pouze informační a není nijak zpracováván.

Následující ukázka představí velmi jednoduchý dokument pro popis vizuálních možností *TeXu*:

```
\chlyph\csaccents
%tohle nastavuje ČESKÉ formátovací styly a znaky
Následující řádky napsal {\TeX} a šlo mu to {\bf výborně}.
Nový odstavec se {\it hezky česky} odsadil. A nakonec taky něco
z matematiky: $\sqrt{x^2}$.
\bye
```

Výstup z takového .tex souboru v příslušném prohlížeči nebo po vtištění bude:²⁷

Následující řádky napsal T_EX a šlo mu to **výborně**.
Nový odstavec se *hezky česky* odsadil. A nakonec taky něco z matematiky:
 $\sqrt{x^2}$.

TeX sám o sobě se nikdy příliš nepoužíval. Až jeho rozšíření přinášejí opravdovou moc pro tvůrce a navíc podstatně zjednodušují tvorbu dokumentu díky možnostem automatizace procesů. Další ukázka se proto bude týkat *LaTeXu*, který jako nástavba *TeXu* nabízí pokročilé funkce. Např. jeho nástroj *BibTeX* umožňuje provádět citace dle vybrané normy, bibliografie, rejstříky, obsahy, poznámky pod čarou, aj.

²⁷ Zde je třeba znovu zdůraznit, že typografický výstup bude mít výrazně vyšší kvalitu, než nabízí program *Microsoft Word*, ve kterém je napsána tato bakalářská práce. Např. odmocnina bude zasahovat po celé délce výrazu, jak je stylisticky správné, kurzíva a středně tučné písmo budou mít elegantnější řez. Starší verze programu *Word* také neumějí slitky, a tak by se dalo pokračovat.

Syntaxe *LaTeXu* vychází ze *Scribu*. Jeho neocenitelnou možností je také programování vlastních maker. Zatímco první ukázka se týkala ovlivnění vzhledu dokumentu, ta následující představuje jeho logickou strukturu. Můžeme tak jasně vidět inspiraci pro budoucí značkovací jazyky:

```
\documentstyle[czech,a4,12pt]{report}
\pagestyle{empty}
\begin{document}
\title{Značkovací jazyky a jejich použití}
\author{Ondřej Smíšek}
\date{14. leden 2011}
\maketitle
...
\end{document}
```

Smyslem *BibTeXu* je oddělení bibliografických informací od jejich formátování. Data jsou umístěna v samostatném databázovém souboru s příponou *.bib*, formát dat je pak definován v dalším souboru *.bst*. Tímto způsobem lze jednoduše měnit citační normy a mít jediný databázový soubor s oborovou literaturou pro všechny dokumenty. Ukázka citace v databázovém souboru:

```
@BOOK{smisek:znackovacil1,
author      = "Ondřej Smíšek",
title       = "Značkovací jazyky a jejich použití",
publisher   = "UISK FF UK",
address     = "U Kříže 8, Praha 5, 15800",
year        = "2011",
note        = "40\,s.", }
```

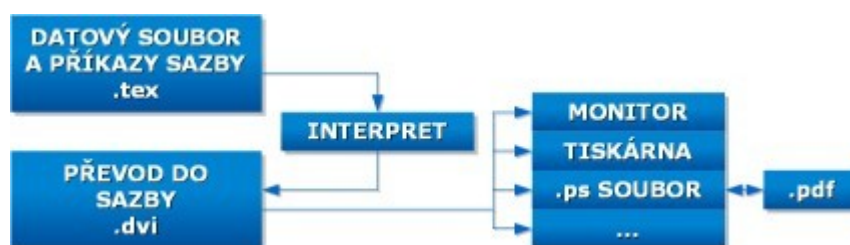
Začlenění citace do dokumentu:

```
Nějaké moudro autora \citep[s.\,19]{smisek:znackovacil1}.
```

Výsledkem bude přímá citace:

Nějaké moudro autora [SMÍŠEK, 2011, s. 19].

Vynikající zprávou je, že *TeX* si rozumí i s *PostScriptem*, modernějším jazykem pro *DTP* především z hlediska grafických možností. Protože mezi *PostScriptem* a *PDF* existují plně funkční převodníky, je *TeX*, díky svým typografickým možnostem, i po více než třiceti letech od uvedení, živým a plně funkčním formátem.



Obr. 6: Zjednodušené schéma práce *TeXu*
[získáno: 03-19-2011; zdroj: autor].

2.6 DTD

Koncept *DTD* vznikl v laboratořích *IBM* roku 1978. V jazycích založených na *SGML*²⁸ a používajících *DTD*, může kodér definovat, jaké prvky a atributy jazyka budou k dispozici a jak je půjde navzájem kombinovat. Tímto způsobem lze „vynalézt“ syntaxi vlastního značkovacího jazyka pro specifické případy použití. *DTD* každopádně nic neříká o významu značek, ten určuje až konkrétní procesor zpracovávající dokument [MIKLE, 2004, s. 133 – 142].

DTD je samostatně stojícím souborem, sám o sobě nic neznamená a nic ho nezpracovává. V zatím představených značkovacích systémech tak zaujímá poněkud zvláštní místo, navíc není pro kodéry a vývojáře důležité ho ovládat, protože – pokud nehodlají definovat vlastní jazyk – s ním nemusejí přijít vůbec do styku.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

V ukázce je předepsaný způsob připojení *DTD* souboru, podle kterého se určí význam značek a jejich atributů v nějakém *XHTML* souboru. Použitá deklarace je navíc striktní, tzn. měla by odpovídat takové verzi *XHTML*, kterou její tvůrci W3C považují za nejvhodnější.²⁹ Konečně také vidíme, že *DTD* soubor je volán absolutně z externího zdroje (server *W3C*), ale možné je i přímé deklarování v dokumentu ve tvaru:

```
<!DOCTYPE kořenový_element [...deklarace_prvků...]>
```

Což může konkrétně vypadat takto:

```
<!DOCTYPE koren SYSTEM "./muj-jazyk.dtd">
```

V praxi je nejvhodnější používání externí deklarace, neboť jeden *DTD* soubor pak sdílí větší množství dokumentů.

DTD nabízí tyto deklarace:

- **ELEMENT** (prvek): definuje značku.
- **ATTLIST** (atributy): definuje atributy značky.
- **ENTITY** (entita): definuje zástupce něčeho (nejčastěji externího souboru).
- **NOTATION** (notace): přiřazuje souborům zpracovatelský program.
- **ENTITY %** (makroentita): definuje pomocnou entitu (zástupce libovolného řetězce v souboru *DTD*, slouží pro zjednodušení zápisu).
- **DOCTYPE** (typ dokumentu): specifikuje odkaz na soubor *DTD* (mnohdy jediné setkání „běžného“ kodéra s jazykem *DTD*).

Před praktickou ukázkou ještě vysvětlení řetězců určujících výskyt značek při deklaraci prvků:

- Znak otazník (?) určuje nepovinný výskyt.
- Znak plus (+) určuje alespoň jeden výskyt.
- Znak hvězdička (*) určuje libovolný počet výskytů.
- Řetězec (EMPTY) představuje prázdnou značku.

²⁸ Mluvíme-li o současnosti, bude to pochopitelně nástupce *SGML* – *XML*, viz další kapitoly.

²⁹ Podmínkového tvaru je v této souvislosti užito záměrně. Webové prohlížeče, pokud nechtějí ztratit své uživatele, se chovají přesně naopak – kombinují a dokáží bez protestů zobrazit i více syntaxí dohromady (např. *HTML* a *XHTML*). Vzhledem k obrovskému množství nesprávně tvořených stránek (ať už z důvodu neznalosti tvůrců nebo nedokonalostí generátorů) by byla většina webu nepoužitelná. Znalost *DTD* je tedy klíčová pro vývojáře browserů.

- Řetězec (ANY) představuje libovolný obsah.
- Řetězec (#PCDATA) představuje pouze data.
- Styl řetězců (a1, a2, a3,...) určuje přesné pořadí.
- Styl řetězců (a1 | a2 | a3 | ...) určuje možnost výběru.

Praktická ukázka *DTD* následuje nyní. Nejdříve samotný syntaktický soubor `motory.dtd`:

```
<!ELEMENT motory      (ANY)>
<!ELEMENT motor       (název, typ, objem, výkon)+>
<!ELEMENT název       (#PCDATA)>
<!ELEMENT typ         (#PCDATA)>
<!ELEMENT objem       (#PCDATA)>
<!ELEMENT výkon       (koní?, kilowatt+)>
<!ELEMENT koní        (#PCDATA)>
<!ELEMENT kilowatt    (#PCDATA)>
```

Následuje *XML* soubor používající právě vytvořené *DTD*:

```
<?xml version="1.0" encoding="windows-1250"?>
<!DOCTYPE motor SYSTEM "motory.dtd">
<motory>
  <motor>
    <název>Magnum</název>
    <typ>V8</typ>
    <objem>7,5</objem>
    <výkon>
      <kilowatt>280</kilowatt>
    </výkon>
  </motor>
  <motor>
    <název>Hemi</název>
    <typ>V8</typ>
    <objem>7,2</objem>
    <výkon>
      <koní>425</koní>
      <kilowatt>315</kilowatt>
    </výkon>
  </motor>
</motory>
```

2.7 Proprietární formáty

Na přelomu 70. a 80. let se mnoho firem vrhlo na oblast *DTP*, přičemž většina z nich existuje dosud. Objevily se volně dostupné jehličkové (1970), laserové (1975) a inkoustové tiskárny (komerčně až po roce 1990³⁰). Výrobci těchto zařízení pro ně připravovali specifické tiskové jazyky. Protože byly v drtivé většině proprietární, nelze o nich v rámci tématu této práce mluvit jako o univerzálních značkovacích (tiskových) jazycích.

Během překotných začátků téměř každá nová tiskárna přinášela svou verzi typografického procesoru. Takové řešení se samozřejmě ukázalo jako neudržitelné. Vše mělo, stejně jako dnes, směřovat k univerzalitě a jedině otevřené systémy mohly v tvrdé konkurenci obstát.

Dále bude telegraficky představeno několik nejvýznamnějších výrobců *DTP* technologií a jejich jazyků pro tiskové úlohy do roku 1990.

- *Canon*: jazyky *CaPSL* (později nahrazen *PCL*) a *UFR* (klon *PCL*).
- *Epson*: jazyk *ESC/P* pro jehličkové a inkoustové tiskárny. Poměrně rozšířený, používali ho i jiní výrobci.

³⁰ Výrobci dlouho bojovali s technologií optimálního nanášení inkoustu na papír.

- *Hewlett-Packard*: Dominantní jazyky pro plottery – *HP-GL*, *HP-GL/2*, *HP-RTL*. Jazyk *PCL* pro tiskárny představen v roce 1984, stal velmi rozšířeným i v konkurenčních zařízeních, největší soupeř *PostScriptu*. Vyvíjen doposud.
- *Xerox PARC*: *InterPress*, alternativa *PostScriptu*.

2.8 PostScript

PostScript vydala firma *Adobe Systems* v roce 1985. Za běžných okolností by nedostal v této práci samostatnou kapitolu, ale své místo by našel mezi proprietárními formáty. Navíc ho nelze řadit mezi značkovací jazyky, ale jednoznačně mezi jazyky programovací, příp. *PDL*. Důvodem, proč stojí samostatně, je skutečnost, že posloužil také pro přípravu a prezentaci internetových dokumentů během 2. pol. 80. let. Proto si na tomto místě zaslouží více pozornosti.

PostScript je postfixový programovací jazyk pro popis tiskového dokumentu, nezávislý na platformě a zařízení. Jeho možnosti lze shrnout do třech oblastí – práce s textem, práce s geometrickými obrazy a práce s importovanou grafikou.

Co se týče textu, tam se neděje nic převratného, co by neposkytovaly již dříve představené systémy. Za zmínku stojí snad jen možnost natočení textových řetězců pod libovolným úhlem. Značné možnosti skýtá geometrické vykreslování. Dokument lze osvěžit vlastními čarami, bublinami, rámečky, tabulkami či prostorovými objekty, přičemž vše se programuje přímo ve zdrojovém souboru, takže není nutné vkládat další externí data. Co se týče vkládané grafiky (bitmapové či vektorové soubory), tu lze dále upravovat – otáčet, měnit velikost – přímo příkazy *PostScriptu*.

PostScript je velmi mocný nástroj a právem si vydobyl přední místo v oblasti tvorby *DTP*, které – spolu se svým nástupcem *PDF* – hájí dosud. Protože tato práce má značně omezený rozsah, budou uvedeny jen dva krátké příklady v pravdě téměř nekonečných možností *PostScriptu*.

Nejprve ukázka zápisu krátkého textu:

```
/Times-Roman findfont
16 scalefont
setfont
72 200 moveto
(Ahoj, světe!) show
showpage
```

Výsledek:

Ahoj, světe!

Druhá ukázka má vykreslit překrývající se různobarevné čtverce:

```
/box
{ 72 0 rlineto
0 72 rlineto
-72 0 rlineto
closepath } def
newpath %černý
252 324 moveto box
0 setgray fill
newpath %šedý
270 360 moveto box
.4 setgray fill
```

```

newpath %světle šedý
288 396 moveto box
.8 setgray fill
showpage

```

Výsledek:



Obr. 7: Obrazce vykreslené *PostScriptem*
[získáno: 02-05-2011; zdroj: autor].

2.8 SGML

Již někdy ve 2. pol. 70. let bylo zřejmé, že univerzalita na poli značkovacích jazyků může velmi usnadnit a urychlit další vývoj počítačových systémů. Jako nejpovolanější se v této oblasti cítila *IBM*. Jazyk *GML* a koncept *DTD* spojila v jeden metajazyk *SGML*, jenž úspěšně získal statut *ISO* standardu. Principy zmíněných jazyků dostatečně ozřejmily kap. 2.3 a 2.6.

Syntaxe je následující:

```

<značka>
  [obsah]
  <vnořená_značka>
    [obsah vnořeného elementu]
  </vnořená_značka>
  [další obsah]
</značka>

```

Vývoj *SGML* probíhal několik let,³¹ přičemž se na něm podílelo *IBM* (Goldfarb jako vedoucí projektu), *ANSI* i lidé kolem *GenCodu*. Tentokrát se tvůrci vůbec nezabývali prezentačními značkami, ale šlo primárně o smysl (strukturu) dokumentu, který bude zachován nezávisle na platformě a aplikaci. *SGML* již netrpělivě očekávaly orgány státní správy, armáda a velké komerční podniky (myšleno v *USA*). Proč tomu tak bylo nejlépe ozřejmí krátká ukázka:

```

<agenti>
  <agent stát="su">
    <přezdívka>Pan Červený</přezdívka>
    <jméno>John</jméno>
    <příjmení>Doe</příjmení>
  </agent>
  <agent stát="cs">
    <přezdívka>Pan Bílý</přezdívka>
    <jméno> Glen</jméno>
    <příjmení>Parks</příjmení>
  </agent>

```

³¹ Goldfarb na své osobní stránce věnované *SGML* (<http://www.sgmlsource.com/>) uvádí, že vývoj začal rok po dokončení *GML* – tedy 1974. Jiné zdroje uvádějí jako počátek vývoje rok 1978. Ukončení projektu se datuje k roku 1991.

</agenti>

S nadsázkou vymyšlená ukázka předvádí specifický značkovací jazyk v pomyslné databázi bezpečnostní organizace, vytvořený pomocí metajazyka *SGML* a formalizovaný definicí *DTD*. Pokyny *DTD* pro validátor³² předepisují vlastnosti správnosti kontrolovaného dokumentu, např. oddělovací symboly značek od textu, case-senzitivitu,³³ délku názvů značek aj.

Jak pokročilé automatizace a účinné správy obrovského množství dat lze s takovou aplikací dosáhnout je zjevné. „Dokonalá obecnost“ stála za vymezením *SGML* jako jazyka nesmírně obtížného a užívaného jen ve velkých korporacích, které mohly zaplatit školené kodéry. To je důvod, proč na web *SGML* neproniklo, ale pouze inspirovalo Berners-Leeho k napsání jednoduššího *HTML*.



Obr. 8: Schéma *SGML* systému
[získáno: 03-19-2011; zdroj: autor].

Ukázka definice entit v *DTD* souboru:

```
<!ENTITY auto01 „Favorit“>
<!ENTITY auto02 SYSTEM „file:///skoda100.txt“>
<!ENTITY % auto03 „Fabia“>
<!ENTITY škodovka „%auto03; nabízí standardní výbavu.“>
```

Ukázka souboru *SGML*:

```
<škodovky>
  <obsah>&auto01; je z roku 1987.</obsah>
  <obsah>Obsah souboru „skoda100.txt“: &auto02;</obsah>
  <obsah>&škodovka;</obsah>
</škodovky>
```

Soubor *SGML* po validaci:

```
<škodovky>
  <obsah>Favorit je z roku 1987.</obsah>
  <obsah>Obsah souboru „skoda100.txt“: 100, 110, 110R</obsah>
  <obsah>Fabia nabízí standardní výbavu.</obsah>
</škodovky>
```

³² Validátor (*parser*) je rutina, která kontroluje dokument podle příslušných pravidel. Její aplikace v *SGML* vychází z praxe programovacích jazyků. Vyhoví-li dokument předpisu pravidel, nazývá se validním.

³³ Je-li jazyk case-senzitivní (*case-sensitive*), pak rozlišuje zápisy malých a velkých písmen. Např. „abc“ a „ABC“ interpretuje jako dva rozdílné řetězce.

RTF

Jazyk *RTF* vytvořila již dobře zavedená společnost *Microsoft*³⁴ v roce 1987 za účelem snadnější výměny naformátovaných textových dokumentů mezi platformami *MS DOS*, *IBM OS/2*, *MS Windows* a *Mac OS*. V současné době ho podporují všechny důležité *PC* kancelářské aplikace. Jeho značky jsou v základním 7bitovém *ASCII* formátu, tudíž je *RTF* soubor čitelný jako prostý text a není závislý na konkrétní aplikaci. Na první pohled se velmi podobá *TeXu*.

Každý dokument začíná zpětným lomítkem (\) a značkou `rtf[číslo_verze]`. Složené závorky ({}) ohraničují atributy skupiny, kterou značka vymezuje. Jeden takový pár závorek povinně uzavírá i celý dokument. Ukázka zápisu dokumentu, jenž sestává z jednoho nadpisu zarovnaného doprostřed stránky a odstavce zarovnaného do bloku:

```
{\rtf1\ansi\deff0 {\fonttbl {\f0 Times;}}
\fs24
{\pard \qc \fs40
Důležité upozornění
\par}
{\pard \qj
Tady je vypsán nějaký text. Bude muset být dostatečně dlouhý, aby se
pak v ukázce mohl zarovnat k oběma okrajům, tzv. {\b do bloku}. Takže
ještě chvíli něco psát. Výborně, to by snad mohlo stačit.
\par}}
```

Výsledek:

Důležité upozornění

Tady je vypsán nějaký text. Bude muset být dostatečně dlouhý, aby se pak v ukázce mohl zarovnat k oběma okrajům, tzv. **do bloku**. Takže ještě chvíli něco psát. Výborně, to by snad mohlo stačit.

³⁴ *RTF* vymyslel tým vývojářů programujících textový procesor *Word*. První implementaci *RTF* obsahoval *Word 3.0*.

3 Značkovací jazyky 1991 – 2011

3.1 HTML

3.1.1 Obecné vlastnosti

HTML je asi neznámějším zástupcem značkovacích jazyků. Díky jednoduchému zápisu, velké zpětné kompatibilitě, nezávislosti na platformě a téměř nulovým požadavkům na vývojový software může každý prezentovat na webu své vlastní jednoduché stránky. Jazyk *HTML* se skládá z prvků, které popisují dokument a zároveň jsou návodem, jak ho zobrazit [MIKLE, 1996, s. 7]. *HTML* je, stejně jako několik dalších jazyků, potomkem metajazyka *SGML*. Schopnosti tohoto předchůdce a propojení s definicemi *DTD* popisují dostatečně kap. 2.6 a 2.8.

Zápis značek (elementů, tagů) má následující syntaxi:

```
<ELEMENT[ ATRIBUT="hodnota"]>[obsah_elementu] [</ELEMENT>]
```

Některé ze značek jsou povinně párové. Tagy lze je typicky libovolně vnořovat. Vyskytují se v zásadě čtyři typy elementů: nepárové bez atributů, párové bez atributů, nepárové s atributy, párové s atributy. Jednotlivým typům odpovídají ukázky:

```
<HR>
<P>Dobrý den.</P>
<IMG SRC="sipka.gif" ALT="Šipka">
<A HREF="http://www.google.cz/">Přejít na Google</A>
```

Poslední tag je tzv. hypertextový odkaz, jenž má formát obvyklého lokátoru:

```
metoda://server[:port]/cesta/soubor[#kotva]
```

První verze *HTML* vznikla v laboratořích *CERN* a okolnosti tohoto vývoje vysvětlily kap. 1.2 a 1.3. Prototyp dokázal zobrazovat jednoduché texty a hypertextovou navigaci. Jako rozhraní v té době sloužil jediný prohlížeč a editor – *WorldWideWeb (Nexus)*. V roce 1993 se objevil další textový browser *Lynx*, který existuje doposud. Prvním grafickým prohlížečem vůbec byl *NCSA Mosaic*.³⁵

3.1.2 HTML 2.0

Rok 1994 přinesl pod taktovkou organizace *IETF* první oficiální specifikaci – *HTML 2.0*. Jednalo o relativně mocný nástroj pro logické zobrazování textu (nadpisy, odstavce, citáty, poznámky, zdrojové kódy...), hypertext, jednoduché formuláře, a také obrázky. Z hlediska sémantiky šlo o nejčistší verzi jazyka, vzhled elementů nešlo téměř vůbec ovlivňovat.

Takto vypadal zápis minimálního validního *HTML 2.0* dokumentu `pokus.htm`.³⁶

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML>
  <HEAD>
```

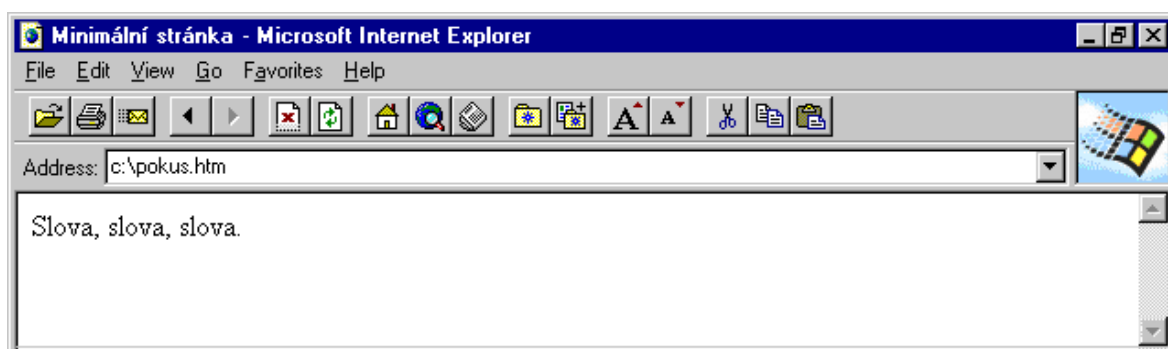
³⁵ Srozumitelně píše o přerodu *HTML* z čistého strukturálního na prezentační jazyk Eric Meyer, světově uznávaný odborník na *HTML* a *CSS*. Meyer si myslí, že právě příchod *Mosaicu* a snaha kodérů a designérů zalíbit se čtenářům také vzhledem stránek, zapříčinila zmatení v zápisech, které trvá víceméně dosud ([16], s. 23 – 25).

³⁶ Starší systémy *MS DOS* a *MS Windows 3.x* nedokázaly pracovat s příponami delšími než 3 znaky. Proto byla zavedena kromě standardní přípony `.html` také `.htm`.


```

<TITLE>
    Minimální stránka
</TITLE>
</HEAD>
<BODY>
    <P>
        Slova, slova, slova.
    </P>
    <!--Komentář, který se nezobrazí-->
</BODY>
</HTML>

```



Obr. 9: HTML dokument „pokus.htm“ zobrazený v *Internet Exploreru 2.0* v prostředí *OS Windows 95* [získáno: 03-18-2011; zdroj: autor].



Obr. 10: Geneze *HTML* v letech 1991 – 2001 [získáno: 03-19-2011; zdroj: autor].

3.1.3 Neformální HTML 3.0

V roce 1995 se strhl velký komerční souboj o nejlepší webový prohlížeč. Soupeřily spolu firmy *Netscape (Navigator)* a *Microsoft (Internet Explorer)*.^{37, 38} Nově vzniklá organizace *W3C* neměla v té době kontrolu nad tím, jaké nestandardní prvky tyto dva soupeři zavádí do svých prohlížečů. Její startovací „čistý“ projekt *HTML+* zcela ztroskotál.^{39, 40} Naopak se objevily blikající a běžící texty, barevné posuvné lišty, přechodové efekty hypertextové

³⁷ Jen v tomto roce vyšly verze *Exploreru 1.0, 2.0* a *3.0*, přičemž každá přinesla zásadní zlepšení. To se zdá proti ustrnutí v prvních letech nového tisíciletí téměř neuvěřitelné.

³⁸ *Microsoft* nakonec zvítězil, *Netscape* dnes řada uživatelů internetu ani nezná. Na přelomu tisíciletí byl *Internet Explorer* nainstalován na více než 90 % osobních počítačů a *Microsoft* s jeho vývojem ustrnul ve verzi 6.0. S příchodem „nové vlny“ prohlížečů (od roku 2004), které nabídly dynamičtější vývoj a pokročilé možnosti prohlížení, interaktivity, správy stahování, záložek a doplňků, podíl *Internet Exploreru* vytrvale klesá. Verze 9.0 uvedená v březnu 2011 dokázala dohnat ztracená léta, otázkou je, zda mu „rozmazlení“ uživatelé ještě dají šanci. Více o problematice vývoje prohlížečů nabízejí *IT* informační servery *Interval* (<http://www.interval.cz/>) nebo *Zdroják* (<http://www.zdrojak.root.cz/>).

³⁹ *HTML+* se mělo stát prvním rozšířením velmi jednoduchého *HTML*. Zavádělo formuláře, tabulky, obrázky s popisky, výpisky, poznámky, podtitulky apod. Zajímavější část z těchto elementů se do pozdějších specifikací nikdy nedostala.

navigace, proprietární meta-tagy a různá další „vylepšení“,⁴¹ která zcela ignorovala snahu o zachování *HTML* jako sémantického jazyka. Konečně, do všeho toho zmatku ještě přišla *Java* se svými applety.⁴²

Během následujícího roku *Netscape* do svého prohlížeče zavedl nové nestandardní prvky jako tabulky, rámy (*frames*),⁴³ obtékání obrázků a matematické vzorce. Pro tato rozšíření se postupně vžilo označení *HTML 3.0*, ačkoliv taková specifikace nikdy nevznikla, a tam, kde jí popisovali, si tak trochu vymýšleli.⁴⁴

3.1.4 HTML 3.2

Ve *W3C* sledovali situaci a uznali smysl některých nových značek. Leden 1997 přinesl oficiální verzi *HTML 3.2*, kde se objevila určitá rozšíření z *Navigatoru* a *Internet Exploreru*, tabulky a obtékání textu. Jazyk se přesto stal nepřehlednou směsicí logických a prezentačních značek. V rozsáhlých projektech, kde měl každý element znovu a znovu definovanou svou podobu, zdrojový kód vypadal jako hrozná směsice znakových řetězců nesoucích minimální význam. A také se tak v zásadě choval – bohužel pro vyhledávače a indexovací stroje.

DTD předpis vypadal následovně:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

Naštěstí už v té době *W3C* pracovalo na jazyku *CSS*. Tato technologie umožňovala zcela oddělit prezentační složku od sémantické struktury dokumentu. Trvalo však řadu let, než se *CSS* plošně zavedlo. Relikty druhé poloviny 90. let lze na webu nalézt doposud.⁴⁵

Toho roku *Microsoft* představil projekt *DHTML* – pojetí každé stránky jako dynamické entity a jejich jednotlivých prvků jako samostatných objektů. Šlo o spojení *HTML*, *DOM*,

⁴⁰ Těch neúspěchů mělo *W3C* v budoucnu ovšem víc. Asi nejkritičtějším obdobím pro tuto jinak velmi důležitou instituci na poli internetových standardů byla léta 2001 – 2007. Tehdy nedokázala vůbec reagovat na potřeby internetové komunity. Více o této malé krizi v oblasti značkovacích jazyků lze nalézt v kap. pojednávající o *XHTML* (viz 3.10). Několik projektů bylo také významně rozpracováno, ale nikdy se nedokončily, resp. nedospěly do stádia doporučení. V době boomu mobilních telefonů to byly jednoduché značkovací jazyky *C-HTML* a *HDML*, založené na *HTML* a přinášející primitivní syntaxi. Oba jazyky nahradilo *WML*, které je v současnosti taktéž překonané. V oblasti stylování *HTML* dokumentů se kromě *CSS* rozvíjel také konkurenční jazyk *JSSS*, se kterým přišla fy *Netscape*. *W3C* upřednostnilo *CSS*, a tak *JSSS* podporoval pouze prohlížeč *Navigator*, a to navíc v jediné verzi. Firmy *Adobe*, *IBM*, *Netscape* a *Sun* definovaly potřebný vektorový jazyk *PGML*. Konsorcium tento návrh zamítlo. O něco úspěšnější byl formát *VML* představený firmami *Autodesk*, *HP*, *Macromedia*, *Microsoft* a *Visio*, nicméně v současnosti ho podporuje pouze *Internet Explorer* a budoucnost je přisuzována formátu *SVG*, který samozřejmě z předchozích dvou zmiňovaných technologií vychází. Neúspěchem *W3C* poslední doby byl formát *XHTML*. S velkou slávou představen v roce 2000, aby byl v roce 2009 po téměř nulovém vývoji zcela zrušen. Velké naděje se nyní vkládají do jazyka *HTML5* (ve spojení se stylovisovou platformou *CSS3*), o němž pojednává poslední část této práce. Oficiální stránky konsorcia jsou zde: <http://www.w3.org/>

⁴¹ „Tím však popřeli základní myšlenku webu, který má umožnit komukoli, odkudkoli a kdykoli snadný přístup k informacím. Interoperabilita webu byla narušena.“ Jiří Kosek v článku pro elektronický časopis *Interval*, více informací zde: <http://interval.cz/clanky/proc-nepouzivam-xhtml/>

⁴² Applet je program v jazyce *Java* obsažený v internetové stránce, typicky spolupracující s *JavaScriptem*. Vkládá se např. mezi proprietární značky `<APPLET></APPLET>`, nebo se načítá z externího souboru.

⁴³ Rámy umožňují v jednom *HTML* souboru zobrazit více dokumentů. Typické použití zahrnovalo jeden rám pro menu stránky a druhý pro samotný obsah jednotlivých odkazů.

⁴⁴ Viz bibliografie [38].

⁴⁵ Detailnější popis *CSS* v této práci je nad její rámec. Je dobré však říci, že *CSS* pracuje na principu dědění vlastností. Je-li například definována barva písma u obecné značky odstavce `<p>`, pak všichni potomci této značky, např. `<p class="potomek">` tuto vlastnost dědí, není-li uvedeno jinak. To umožňuje krátké, přehledné a silné prezentační zápisy. Dle autorova názoru je jednou z nejlepších knih o *CSS* kompletní průvodce Erica Meyera [16].

CSS a skriptovacích jazyků.⁴⁶ Web mohl nastoupit novou cestu efektivní a zároveň efektní prezentace obsahu. Nicméně často z toho vznikala kýčovitá, nepřehledná koláž, zbytečně zatěžující procesor a grafický subsystém počítačů.

3.1.5 HTML 4.0

V prosinci 1997 uvedlo konsorcium novou specifikaci, která chtěla vnést do roztržitého světa *HTML* nový řád, situaci ale naopak dále zamotala. Kvůli použitelnosti starších stránek se zavedly rovnou 3 *DTD* předpisy: striktní, plně kompatibilní s *HTML 3.2* a povolující pouze rámy:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
```

Striktní deklarace umožňovala pro definici vzhledu použít jen *CSS*. Posvěceny byly nové elementy (<OBJECT> pro multimédia), překonané naopak zavrženy (<APPLET>, , <BASEFONT>, <U>, <CENTER> a další). Rozšíření se dočkaly formuláře, zavedly se entity pro znaky z národních abeced, přibyla možnost obousměrného zápisu textu. Lepší podporu dostalo klientské skriptování.

Nastal bohužel očekávaný výsledek: prohlížeče zavedly nové prvky, podporu zavržených elementů zachovaly a kodéři stránek používali přechodové *DTD*.

3.1.6 HTML 4.01

Po dvou letech od vydání *HTML 4.0* přišla revize *4.01* opravující drobné chyby. Tím byl vývoj *HTML* na následující téměř desetiletí ukončen.

Následující ukázka předvede možnosti typografických a sémantických možností úprav textu v *HTML 4.01*. Nejprve zdrojový kód:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<HTML>
  <HEAD>
    <TITLE>
      Nepodstatný název
    </TITLE>
  </HEAD>
  <BODY>
    <h4>Vybrané formátovací značky (většinou zavržené):</h4>
    <p><b>tučné</b>, <i>kurzíva</i>, <b><i>tučná kurzíva</i></b>,
    <u>podtržené</u>, <tt>neproporcionální</tt>, <big>velké</big>,
    <small>malé</small></p><h4>Velikost pomocí elementu FONT
    (zavrženo):</h4><p><font size=1>1</font>, <font size=2>2</font>,
    <font size=3>3</font>, <font size=4>4</font>, <font
    size=5>5</font>, <font size=6>6</font>, <font
    size=7>7</font></p><blockquote>Ukázka formátování
    citátu</blockquote><h4>Vybrané sémantické značky
    (doporučené):</h4><p><q>Přímá řeč v uvozovkách.</q>,
    <em>zvýraznění</em>, <strong>velké zvýraznění</strong>,
    <cite>citace</cite>, <code>programový kód</code>, <var>proměnná
    programu</var>, <abbr>zažítá zkratka, např. WWW</abbr>,
    <acronym>akronym - zkratka která se čte jako
    slovo</acronym></p><h4>Indexy:</h4><p>H<sub>2</sub></p>O,
    E=mc<sup>2</sup></p>
```

⁴⁶ Nejznámějším zástupcem klientského skriptovacího jazyka (takového, který pracuje na uživatelském počítači a nepřistupuje na server) je *JavaScript*. *Microsoft* ovšem zavedl ještě další dva vlastní jazyky: *JScript* (téměř totožný s *JavaScriptem*) a *VBScript* (syntaxe *Visual Basicu*). Oba se dnes na webu prakticky nepoužívají, podpora ze strany jiných prohlížečů než *Internet Exploreru* je podle všeho nulová.

```
</BODY>
</HTML>
```

Výsledný dokument v prohlížeči:

Vybrané formátovací značky (většinou zavřížené):

tučné, *kurzíva*, ***tučná kurzíva***, podtržené, neproporcionální, **velké**, malé

Velikost pomocí elementu FONT (zavříženo):

1, 2, 3, 4, 5, 6, 7

Ukázka formátování citátu

Vybrané sémantické značky (doporučené):

“Přímá řeč v uvozovkách.”, *zvýraznění*, **velké zvýraznění**, *citace*,
programový kód, *proměnná programu*, *zažítá zkratka*, např. WWW,
akronym - zkratka která se čte jako slovo

Indexy:

H₂O, E=mc²

Obr. 11: Zobrazení vybraných textových elementů v internetovém prohlížeči *Mozilla Firefox 3.6*
[získáno: 03-24-2011; zdroj: autor].

3.2 DocBook

Jedním z projektů postavených na základě metajazyka *SGML* byl také *DocBook*. Po dostatečném rozšíření metajazyka *XML* (viz kap. 3.4) byl *DocBook* aktualizován také na jeho základě.

DocBook, rozvíjející se od roku 1991, zpočátku používalo jen několik technologických firem pro zápis dokumentací ke svému softwaru či hardwaru, průkopníkem na tomto poli bylo dále nakladatelství *O'Reilly*. Smyslem vzniku *DocBooku* bylo, aby se při změnách, které jsou v oblasti počítačových produktů velmi časté, mohl automatizovat a zjednodušit proces zpracování a aktualizace dokumentace.

Obliba *DocBooku* rychle vzrostla, protože kdo ovládal *HTML* pro *WWW*, velmi rychle si osvojil i tagy pro sázecí tvorbu. Výhody nezávislosti na platformě a zařízení společně moderním značkovacím jazykům již byly dostatečně ozřejměny. Výstup z *DocBooku* lze rychle formátovat pro různorodé potřeby – k tisku, na internet, či jako nápovědu. Značné oblibě se formát těší mezi vývojáři *open-source* projektů.

Od roku 1999 *DocBook* rozvíjí sdružení *OASIS*⁴⁷ (dříve *SGML Open*), které např. stojí za kancelářskými XML formáty *OpenDocument*.⁴⁸ V roce 2009 vznikla zatím poslední verze 5.0. Nejznámějším propagátorem *DocBooku* v českém prostředí je Jiří Kosek.⁴⁹

DocBook je živou a plně funkční platformou, i když o něm není mnoho slyšet.

Následuje ukázka tohoto jazyka:

```
<?xml version="1.0" encoding="UTF-8"?>
<book xml:id="simple_book" xmlns="http://docbook.org/ns/docbook"
version="5.0">
  <title>Značkovací jazyky a jejich použití</title>
  <chapter xml:id="chapter_1">
    <title>Úvod</title>
    <para>První odstavec.</para>
    <para>Druhý je <emphasis>důležitější</emphasis>.</para>
  </chapter>
  <chapter xml:id="chapter_2">
    <title>Závěr</title>
    <para>A to je konec.</para>
  </chapter>
</book>
```

3.3 VRML

Již počátkem roku 1994 započaly vážné diskuse na téma popisování prostorových objektů a virtuální reality na webu. Nový jazyk měly podporovat prohlížeče stejně dobře jako *HTML* a k pohybu mezi scénami⁵⁰ měla sloužit hypertextová navigace. Zdálo se zbytečné začínat od nuly, a tak vznikl projekt *VRML* vycházející z existujícího formátu *Open Inventor* (*OpenGL*) firmy *Silicon Graphic*.

Konkrétní výsledky na sebe nenechaly dlouho čekat, nicméně zpočátku šlo vytvářet pouze statické scény. V roce 1997 vznikla finální specifikace *VRML 97*, která se stala normou *ISO*.⁵¹ Světy existovaly samostatně nebo jako součást *HTML* dokumentů. Objekty ve scénách se mohly nejen pohybovat, ale také reagovaly na podněty uživatele.

Ve své době se jednalo o dobrou technologii, zvláště na tak nekonzistentní platformě, jakou byl web a jeho prohlížeče. *VRML* nepožadovalo konkrétní systém, jeho ovládání bylo přirozené a jazyk srozumitelný, jak dokládá následující ukázka základní logické struktury:

```
#VRML V2.0 utf8
# Hlavička
WorldInfo { ... }
Viewpoint { ... }
# Všeobecné informace o světě
Transform { ... }
Group { ... }
PositionInterpolator { ... }
# Popis těles, vlastností, definice prvků
ROUTE ... TO ...
# Propojení prvků
```

⁴⁷ Oficiální stránky konsorcia zde: <http://www.oasis-open.org/>

⁴⁸ Dnes velmi silný konkurent kancelářského balíku *Microsoft Office* je právě *OpenOffice*. Je zcela zdarma, podporuje moderní formáty, export do *PDF*, je rychlý a stabilní. Více informací na oficiálním českém portálu zde: <http://www.openoffice.cz/>

⁴⁹ Více informací zde: <http://www.docbook.cz/>

⁵⁰ *VRML* nazývá jednotlivé scény jako světy (*worlds*). Tomu odpovídá i zavedená přípona *.wrl*.

⁵¹ Revize normy proběhla ještě v roce 2004.

Některé kritické nevýhody nakonec tuto technologii odstavily na slepou kolej. Hlavní překážkou masového rozšíření byla obrovská náročnost na výpočetní výkon, světy popisující např. část města s pohybujícími se objekty v podstatě nešlo rozumně zobrazit. Webové prohlížeče neintegrovaly *VRML* přímo, ale používaly externí aplikace, což utvořilo zřejmou překážku většímu rozšíření. Dále se nepočítalo s možností zobrazení *HTML* stránek uvnitř *VRML* světů a jejich vzájemného ovlivnění. Při dnešní detailnosti prostorových scén (rozuměj počtu vrcholů objektů, tzv. polygonů) by navíc textový zápis zabíral neúměrné množství paměti. *VRML* je dnes přes své zajímavé možnosti v podstatě mrtvou platformou. Jeho nástupcem je formát *X3D*, který používá *XML* syntaxi.

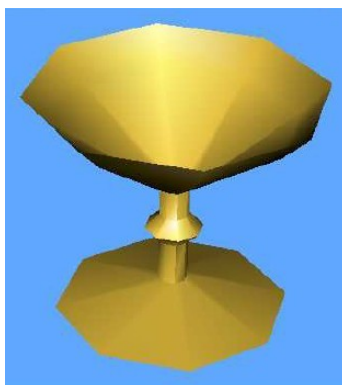


Obr. 12: Logo *VRML 97*. „M“ v akronymu nejprve znamenalo „markup“, později „modeling“
[získáno: 03-20-2011; zdroj: <http://www.web3d.org/>].

Ukázka *VRML*:

```
#VRML V2.0 utf8
Transform {
  children Shape {
    appearance Appearance {
      material Material {
        diffuseColor 1 0.8 0.2
        specularColor 1 1 1 } }
    geometry Extrusion {
      crossSection [ 1 0, 0.7 0.7,
        0 1, -0.7 0.7,
        -1 0, -0.7 -0.7,
        0 -1, 0.7 -0.7,
        1 0]
      spine[ 0 0 0,
        0 0.2 0,
        0 0.5 0,
        0 0.6 0,
        0 0.7 0,
        0 0.9 0,
        0 1.4 0,
        0 1.6 0]
      scale[ 0.8 0.8
        0.1 0.1,
        0.1 0.1,
        0.2 0.2,
        0.1 0.1,
        0.1 0.1,
        0.8 0.8,
        1 1]
      solid FALSE
      endCap FALSE
      creaseAngle 0.78 } } }
```

Výsledek v prohlížeči:



Obr. 13: Pohár vykreslený pomocí VRML
[získáno: 03-18-2011; zdroj: [36]].

3.4 XML

Vzniku jazyka *XML*⁵² předcházely dvě důležité technologie – *HyTime*⁵³ a *DSSSL*.⁵⁴ *XML* je zpřesněnou, zpřísněnou, ale také praktičtější modernější aplikací mohutného *SGML*. Je to otevřený značkovací jazyk pro nejširší použití, je multiplatformní a multilingvální. Slouží jako výměnný formát dat na internetu, jako syntaktická základna sémantického webu⁵⁵ či jako databázový model [MLÝNKOVÁ, I...et al., 2006, s. 7]. Jeho nejsilnějšími zbraněmi jsou možnost definice vlastních značek a naprostá univerzálnost. Práce s *DTD* již byla dostatečně popsána výše (kap. 2.6) a v tomto ohledu *XML* zachovává plně kompatibilní postupy s *SGML* (kap. 2.8). Syntaxe *XML* je popsána v kap. 3.9.

Na *XML* začalo pracovat sdružení *W3C* v roce 1996, verze *1.0* nabyla statutu doporučení počátkem roku 1998, zatím poslední verze *1.1* pak v roce 2004. Hned po svém uvedení se *XML* stalo široce užívanou a podporovanou technologií, protože svět *IT* jej jako nástupce komplikovaného *SGML* velmi očekával.

Obecné vlastnosti *XML* v bodech jsou následující:

- Ideální jazyk pro strukturalizaci dokumentů, elementům lze přiřadit význam
- Zápis v podobě běžného textu, kódování *Unicode*
- Pochopitelná syntaxe založená na *HTML*
- Žádná omezení platformami, licencemi, jazyky
- Nízké požadavky na software v závislosti na zkušenostech kodéra
- Snadná (automatická) konverze do jiných formátů
- Hypertextová navigace a odkazy

⁵² Na *ÚISK* psal v r. 2003 o *XML* např. J. Kaplický rigorózní práci s názvem „Jazyk XML a jeho uplatnění v databázových aplikacích“. Více informací zde: <http://isdp.alstanet.cz/PublicationView.aspx?PublicationID=402>

⁵³ Hypertextový značkovací jazyk, jehož tvůrcem byl v této práci již několikrát zmiňovaný Ch. Goldfarb. *HyTime* představoval první konkrétní aplikaci *SGML*, v roce 1992 se stal také *ISO* normou. Je předchůdcem *HTML* a *XML*.

⁵⁴ *SGML* stylový jazyk, používá ho např. *DocBook*. Přímý (a sofistikovanější) předchůdce *CSS* a *XSL*.

⁵⁵ Hlubší představení sémantiky, její směřování a používané technologie, není v kompetenci této práce. O tématu bylo na *ÚISK* psáno již několikrát. V r. 2005 obhájil O. Horsák diplomovou práci s názvem „Struktura RDF pro metadata, její vývoj a perspektivy aplikace v digitálních systémech“. V r. 2009 obhájili M. Zimová a E. Piňos diplomové práce „Vývoj sémantického webu“ a „Sémantický web – ontologický Internet budoucnosti“. Více informací zde: <http://isdp.alstanet.cz/PublicationView.aspx?PublicationID=1112> a zde: <http://isdp.alstanet.cz/PublicationView.aspx?PublicationID=951>

- Modularita: *XML* popisuje jen tagy a jejich atributy. Na něm jsou pak postaveny konkrétní aplikace⁵⁶
- Zdrojový kód je relativně dlouhý, ale lze ho komprimovat

Čím lépe je dokument sémanticky strukturován, tím jednodušší je ho zpracovávat, uchovávat a transformovat. Životnost takového dokumentu je řádově vyšší, teoreticky neomezená. Míra entropie⁵⁷ je velmi nízká, hodnota informace naopak vysoká. Toto splňují metajazyky *SGML* a *XML*.



Obr. 14: Porovnání některých dokumentových formátů z hlediska míry informací
[získáno: 04-03-2011; zdroj: autor].

3.5 WML

S rozvojem mobilní komunikace vznikla také snaha o zpřístupnění internetu přes mobilní telefony. V roce 1998 organizace *Wap Forum* definovala systém *WAP* jako *GSM* ekvivalent internetových protokolů. Jazyk pro *WAP* se jmenoval *WML* a představoval jednoduchou implementaci *XML*. V první verzi nepodporoval barvy a pro obrázky byl určen monochromatický 1bitový formát *WBMP*.



Obr. 15: Obrázek ve formátu *WBMP*
[získáno: 04-29-2011; zdroj: autor].

⁵⁶ V případě *XML* je dobré zmínit následující nejrozšířenější aplikace: *Xlink* popisuje hypertextové odkazy. *DOM* pracuje s elementy dokumentu jako se samostatnými objekty, čehož se využívá k jejich obsluze (např. *JavaScriptem*). Na *DOM* navazují „jmenné prostory“ (*Namespaces*), jejichž úkolem je umožnit používat v jednom dokumentu více značek pro elementy a atributy definované v různých schématech. Může jít např. o kombinaci *XHTML* (vlastní webová stránka), *SVG* (vektorová mapa) a *RDF* (metadata). *Xpath* prohledává a adresuje jednotlivé části *XML* dokumentu. Jeho doplňkem je *XPointer* sloužící k odkazování na části dokumentů (podobné jako kotvy v *HTML*). *CSS* lze aplikovat na *XML* podobně jako na *HTML*, nicméně spíše se používá *XSLT*.

⁵⁷ Entropie (neuspořádanost) je princip původem z termodynamiky, za její přenos do oblasti informační teorie se vděčí matematikovi C. Shannonovi. Snižování entropie v nějakém systému má za následek zvyšování jeho informační hodnoty, ovšem za cenu zvýšené nestability. Typický *XML* dokument je velmi „nestabilní“, plný komplikovaných značek a zápisů, nicméně velmi informačně nabytý. Systém s vysokou mírou entropie je stabilní a klidný, ale informačně nezajímavý, např. na tabuli napsaná řada stejných znaků bez zjevného smyslu.

Původní *WML* bylo asi po třech letech nahrazeno zjednodušeným *XHTML* (tzv. *XHTML Mobile Profile* nebo nepřesně *WAP 2.0*) s podporou barev, obvyklých grafických formátů (*GIF*, *PNG*, *JPG*), tabulek a *CSS*.

Dnešní *smart-phones* dokáží interpretovat běžné *HTML* a *XHTML* *WWW* stránky. *WAP* jako takový pro svou nízkou grafickou atraktivitu postupně zaniká.

Ukázka *WAP* souboru:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card>
    <p>Tento text bude zobrazen na displeji telefonu.</p>
  </card>
</wml>
```

3.6 SVG

V 80. a 90. letech existovalo již dostatečné množství bitmapových grafických formátů (*GIF*, *JPG*, *PNG*), které se samozřejmě rozšířily i na internetu, potažmo webu. Po dlouhá léta však nebyl k dispozici obecně použitelný (a také otevřený) formát pro vektorovou grafiku.⁵⁸ Webové stránky ze své podstaty⁵⁹ přímo volaly po pružně se přizpůsobujícím grafickém výstupu, proto v roce 1998 vznikla iniciativa *SVG*.

SVG je značkovací jazyk pro popis dvourozměrné vektorové grafiky založený na *XML* syntaxi, který je dostatečně přehledný, výkonný a funkční – spolupracuje např. s *CSS* a *DOM*. Od roku 2005 se již prakticky nevyvíjí, ale integrace do webových prohlížečů probíhala překvapivě pomalu, nutností byly samostatné utility [HEROUT, 2010, str. 248 – 249]. Ještě v roce 2010 byla částečná, letos však byla dovršena.⁶⁰

SVG používá k vykreslování tzv. cesty (*paths*), kde se vypíše průchod od počátečního ke koncovému bodu objektu. Pro zjednodušení definuje také následující „hotové“ vektorové tvary: čtyřúhelník, kružnici, elipsu, úsečku, lomenou čáru, mnohoúhelník a křivku. Dále lze importovat klasickou rastrovou grafiku a texty. To vše se v *SVG* stává uzavřeným, samostatným objektem. Tyto objekty lze seskupovat, překrývat, polohovat, ořezávat, zprůhledňovat, stínovat či dokonce animovat. Ne všechny tyto možnosti ale mohou podporovat současná zařízení. Proto existují tři úrovně implementace *SVG*: *Full* (plná podpora vlastností; pro *PC*, tablety apod.), *Basic* (omezené filtry; pro *PDA*, *smart-phones* apod.) a *Tiny* (bez skriptů, filtrů, barevných přechodů, průhlednosti a *CSS* formátování; pro mobilní telefony).

Formát má díky svým vlastnostem nakročeno ke slibné budoucnosti. Lze ho úspěšně komprimovat, generovat skripty, může být samostatným souborem nebo vložen do *HTML* či *XML* stránky, nebo do jiného *SVG* souboru. Může být importován a exportován velkým množstvím moderních kancelářských a grafických programů. Své místo může nalézt nejen na webu, ale také ve sféře průmyslu pro technické dokumentace (např. jako náhrada *DXF* fy *AutoDesk*), či obecně jako univerzální formát pro přenos vektorových informací.

⁵⁸ Sluší se zmínit možnosti jazyka *VRML*, ale to bylo pro případ jednoduchých obrázků či tabulek na webu příliš komplikované řešení (více kap. 3.3). Další možností je použití *PostScriptu*, který ale opět nelze zpracovat přímo a je značně syntakticky složitý. Samostatnou kapitolou je oblíbený *Flash*, jenž je účinným, nicméně uzavřeným formátem.

⁵⁹ Standardně nevíme, jaké zařízení a s jak velkou obrazovkou koncový uživatel použije.

⁶⁰ *SVG* již podporuje pět nejdůležitějších webových prohlížečů.

Následuje jednoduchá ukázka, jenž má vykreslit dva obrazce. Nejprve zdrojový kód:

```
<?xml version="1.0"?>
<svg xmlns="http://www.w3.org/2000/svg">
  <title>Kreslení pomocí SVG</title>
  <polygon style="fill:green;stroke:red;stroke-width:5"
    points="350, 75 379,161 469,161 397,215
           423,301 350,250 277,301 303,215
           231,161 321,161" />
  <polygon style="fill:grey;stroke:black;stroke-width:10"
    points="850, 75 958,137.5 958,262.5
           850,325 742,262.6 742,137.5" />
</svg>
```

A výsledek, který bude stejný všude tam, kde je *SVG* podporováno – v prohlížeči, grafickém editoru či textovém procesoru:



Obr. 16: Grafika vykreslená pomocí *SVG*
[získáno: 03-17-2011; zdroj: autor].

3.7 SMIL

Pro multimediální prezentace vytvořilo *W3C* jazyk *SMIL*, jenž vychází ze syntaxe *XML*. Dokument může obsahovat text, obrázky, hypertextové odkazy na další prezentace, přehrávat video, zvuky apod. Na první pohled se velmi podobá tradičnímu *HTML*.

První verze se objevila záhy po vydání doporučení *XML* v roce 1998. Verze 2.0 z roku 2001 přinesla lepší provázání s dalšími jazyky, např. *SVG*. Rok 2005 přinesl verzi 2.1, rok 2008 zatím poslední verzi 3.0.

SMIL spíše než na internetu, kde je jeho podpora komplikovaná, nachází své využití na různých proprietárních zařízeních, která nejsou provázána se stávajícími technologiemi, na tabletech, při ovládání video disků či na multimediálních reklamních plochách.

Struktura zápisu je následující:

```
<smil>
  <head>
    <meta name="copyright" content="Your Name" />
    <layout>
      <!-- layout tags -->
    </layout>
  </head>
  <body>
    <!-- media and synchronization tags -->
  </body>
</smil>
```

Konkrétní ukázka jazyka není dle autora nutná. Na internetu se *SMIL* běžně neobjevuje, pro typické uživatele je to v zásadě neznámá odnož *XML*. Syntaxe se podobá kombinaci *HTML*, *SVG* a *CSS*.

3.8 MathML

Podmnožinou jazyka *XML*, čímž je v oblasti počítačového značkování v posledních letech takřka všechno, je také jazyk *MathML*. Samotný název plně vystihuje jeho účel: je primárně určen k zápisu matematických výrazů a vzorců. První verze byla uvolněna v roce 1999, v roce 2010 nabyla statutu doporučení poslední verze 3.0. Ta je pravděpodobně také verzí poslední. Vývoj koordinuje *W3C*.

Potřeba jazyka pro zápis matematických vzorců na *WWW* je nepochybně stejně stará jako web samotný, dlouho však neexistovalo odpovídající řešení této situace. Do jisté míry se opakovala situace ze 70. let, kdy D. Knuth z nespokojenosti se stavem počítačové typografie a jejího přístupu k matematice, vymyslel vlastní úspěšný program *TeX* (kap. 2.5).

První verze *HTML* byla omezená ve všem, matematické zápisy neumožňovala vůbec. Postupně se implementovaly speciální entity, horní a dolní indexy, to však stále nestačilo. Pomocí *HTML* takovýto vzorec prostě nešlo a nelze zapsat:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Obr. 17: Vzorec řešení kvadratické rovnice
[získáno: 25-04-2011; zdroj: autor].

Jediným řešením tak bylo vzorce vkládat jako obrázky. To s sebou neslo mnoho nevýhod: velký datový objem souborů, nutnost programů, které obrázky umějí vytvořit, nulová informační hodnota obrázků pro indexovací roboty, nemožnost dynamické aktualizace dat apod.

MathML popisuje primárně význam jednotlivých matematických značek, ale zároveň i jejich vzhled, pokud není přiložen speciální styl, protože tyto symboly mají svou typickou podobu. Zápis v *MathML* je na první pohled dlouhý a komplikovaný; nepředpokládá se, že ho bude tvořit člověk ručně, jako je to typické u *TeXu*, ale nějaká *XML* aplikace. To působilo zpočátku trochu problémy, v současné době již existuje mnoho převodních utilit, dobrá podpora je v kancelářských programech a *WWW* prohlížečích.

Nyní následují ukázky kódů vzorce řešení kvadratické rovnice z obr. 17. V prvním případě jednoduchý zápis v *TeXu*, kde se čekává přímá editace člověkem:

```
x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
```

A v *MathML*:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mi>x</mi>
    <mo>=</mo>
    <mfrac>
      <mrow>
        <mrow>
          <mo>-</mo>
          <mi>b</mi>
        </mrow>
        <mo>±</mo>
        <msqrt>
          <mrow>
```

```

<msup>
  <mi>b</mi>
  <mn>2</mn>
</msup>
<mo>-</mo>
<mrow>
  <mn>4</mn>
  <mo>&#x2062;</mo>
  <mi>a</mi>
  <mo>&#x2062;</mo>
  <mi>c</mi>
</mrow>
</mrow>
</msqrt>
</mrow>
<mrow>
  <mn>2</mn>
  <mo>&#x2062;</mo>
  <mi>a</mi>
</mrow>
</mfrac>
</mrow>

```

3.9 RSS

V roce 1999 společnost *Netscape* uvedla přínosnou novinku, kterou použila na své *home-page*. Jednalo se o stránku, která zobrazovala a v určité časové periodě obnovovala nově vydané články. Uživatelé díky tomu nemuseli složité prohledávat celý web – stačilo jim přidat si tuto stránku, tzv. *RSS* čtečku („*RDF Site Summary*“ *reader*, *feed* nebo *channel*) mezi oblíbené položky a měli automaticky aktualizovaný přehled všeho, co se na webu v posledních hodinách objevilo.



Obr. 18: Logo *RSS*

[získáno: 28-04-2011; zdroj: Příspěvatelé Wikipedie. Dostupné z *WWW*: <http://cs.wikipedia.org/>].

Technologie, založená na v té době ještě nedokončeném projektu *RDF*⁶¹, rychle získávala na oblibě především u velkých zpravodajských serverů. Během tří let prodělalo *RSS* bouřlivý vývoj; v roce 2002 vznikla v současnosti nejrozšířenější verze *2.0*, v roce 2005 *3.0 Lite*. Ve verzích sice panuje poněkud chaos, důležitá je však skutečnost, že *RSS* je aplikací metajazyka *XML*, což znamená jednoduchou implementaci a širokou podporu napříč servery i prohlížeči.

První verze *RSS* toho mnoho nenabízela, byl to jednoduchý výpis článků. Následuje ukázka zdroje s jedním článkem:

⁶¹ Jak bylo řečeno výše, o sémantice internetových dokumentů či aplikací již na *ÚISK* vzniklo několik prací. V této práci proto nebudou technologie jako *RDF*, *RDFa*, mikroformáty či *OWL* znovu popisovány. V souvislosti s tématem se však autorovi zdá přínosné zmínit alespoň dvě zajímavé diplomové práce. V r. 2008 N. Pleska obhájila práci s názvem „Web 2.0 a blogy a jejich význam pro rozvoj internetové komunikace“. Více informací zde: <http://isd.p.alstanet.cz/PublicationView.aspx?PublicationID=90>. V r. 2010 pak E. Hutníková obhájila velmi aktuální práci s názvem „Mashupy a kultura API“. Více informací zde: <http://isd.p.alstanet.cz/PublicationView.aspx?PublicationID=1032>

```

<?xml version="1.0" encoding="windows-1250"?>
<!DOCTYPE rss PUBLIC "-//Netscape Communications//DTD RSS 0.91//EN"
"http://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
  <channel>
    <title>Superserver.com - články</title>
    <link>./clanky/</link>
    <description>informace nejen ze světa</description>
    <language>cs</language>
    <item>
      <title>Lorem ipsum dolor (28. 4. 2011)</title>
      <link>/clanky/lorem-ipsum-dolor-28-4-2011/</link>
      <description>Anotace článku. Tralala...</description>
    </item>
  </channel>
</rss>

```

V prohlížeči, který RSS neumí zpracovat, ale podle hlavičky rozpozná, že jde o XML dokument, se ukázka zobrazí nějak takto:

```

- <rss version="0.91">
- <channel>
  <title>Superserver.com - články</title>
  <link>./clanky/</link>
  <description>informace nejen ze světa</description>
  <language>cs</language>
- <item>
  <title>Lorem ipsum dolor (28. 4. 2011)</title>
  <link>/clanky/lorem-ipsum-dolor-28-4-2011/</link>
  <description>Anotace článku. Tralala...</description>
</item>
</channel>
</rss>

```

Obr. 19: XML dokument v prohlížeči Mozilla Firefox
[získáno: 28-04-2011; zdroj: autor].

V prohlížeči Opera se RSS čtečka zobrazí následujícím způsobem:



Lorem ipsum dolor (28. 4. 2011)

28.4.2011 12:06:18

Anotace článku. Tralala...

Obr. 20: Standardní vzhled RSS kanálu v prohlížeči Opera
[získáno: 28-04-2011; zdroj: autor].

RSS 2.0 přišlo s několika vylepšeními, přibýly nové elementy pro upřesnění článků, jednoznačná identifikace, zrušení omezení počtu článků (elementů `<item>`) apod. Více samotná ukázka:

```
<?xml version="1.0" encoding="windows-1250" ?>
<rss version="2.0">
  <channel>
    <title>RSS čtečka</title>
    <description>Ukázková čtečka</description>
    <link>http://www.example.com/rss.html</link>
    <lastBuildDate>Mon, 06 Sep 2010 00:01:00 +0000</lastBuildDate>
    <pubDate>Mon, 06 Sep 2009 16:45:00 +0000</pubDate>
    <item>
      <title>Ukázkový článek</title>
      <description>Nějaký text, tralala...</description>
      <link>http://www.example.com/</link>
      <guid>identifikátor, např. URI</guid>
      <pubDate>Mon, 06 Sep 2009 16:45:00 +0000</pubDate>
    </item>
  </channel>
</rss>
```

3.10 XHTML

První oficiální specifikace jazyka *XHTML* vyšla počátkem roku 2000. *XHTML W3C* zamýšlelo jako nástupce *HTML*, které mělo postupně nahradit. Na první pohled se od sebe tyto jazyky téměř nelišily a řada lidí z oboru se ptala, jaký byl vlastně smysl pro vytvoření a propagování něčeho tak marginálně rozdílného.

Hlavní rozdíl mezi *HTML* a *XHTML* byl v syntaxi – *SGML* versus *XML*. *XHTML* dokument byl díky tomu jednoduše zpracovatelný pomocí *XML* nástrojů a převoditelný do jiných formátů. Zavrženy byly další nesémantické značky a vyžadováno používání *CSS* pro vzhled. Měl to být jazyk čistějších, jednodušších a praktičtějších webových stránek.

Striktní *XML* syntaxe přinesla množství nových problémů, které vývojáři nebyli zvyklí řešit, nebo je, bohužel, řešit nechtěli:

- Elementy se nesmí křížit
- Case-sensitivita – názvy elementů a atributů musí být napsány malými písmeny
- Pro všechny elementy jsou vyžadovány ukončovací značky (např. `
`)
- Hodnoty všech atributů musí být v uvozovkách
- Minimalizace atributů není dovolena
- Některé elementy se nemohou více zanořovat (např. `<form>`)
- Povinná úniková sekvence pro řídicí znaky (`<![CDATA[&, <, >]]>`)

Tvůrci browserů se pochopitelně – aby nepřišli o uživatele – zachovali tak, že podporovali nové, ale i zakázané elementy a formulace.⁶² Do karet jim hrálo i opětovné třívrstvé *DTD* (*Strict*, *Transitional*, *Frameset*). Bylo jedno, zda jsou v kódu *HTML* nebo *XHTML* značky najednou. Tvůrce netrápil zápis, a tak jim „nový“ jazyk nepřinášel žádný užitek.

V roce 2001 vyšla revize *XHTML 1.1* definující moduly pro různé potřeby (*XHTML Print* pro tisk, *XHTML Basic* pro mobilní zařízení), ale opět to nebyl zvláštní posun. *XHTML* ni-

⁶² V nejkritičtější době navíc nejrozšířenější prohlížeč *Internet Explorer* neuměl korektně zpracovat *XHTML* deklaraci (nerozuměl *XML* hlavičce a zároveň *HTML* zápisu). Tudiž *XHTML* stránky vůbec nezobrazil nebo s obtížemi. Konzervativní a firemní zákazníci si tak udělali na *XHTML* negativní názor (pokud vůbec nějaký).

kdo nepotřeboval. Jazyk osm let stagnoval a ztrácel přívržence, aby byl roku 2009 jeho vývoj oficiálně ukončen v rozpracované fázi *XHTML 2.0*.⁶³

Ukázka striktního *XHTML* dokumentu:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <title>A XHTML document</title>
  </head>
  <body>
    <p>Ukázkový odstavec<br />a nový řádek.</p>
    
  </body>
</html>
```

3.11 Atom

Jistou evoluci, nikoli revoluci, představoval formát *Atom*, uvedený v roce 2003 a myšlený jako nástupce *RSS*.⁶⁴ Atom kladl důraz na kvalitnější návrh standartu a otevřenost. Verze 0.3 se rychle rozšířila, protože zaujala *Google*, který jej začlenil do svých aplikací. V roce 2005 vznikla finální verze 1.0. Obě technologie agregace a syndikace⁶⁵ dat dnes fungují paralelně.

Ukázka formátu *Atom 1.0*:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>Název</title>
  <subtitle>Podnázev</subtitle>
  <link href="http://example.org/feed/" rel="self"/>
  <link href="http://example.org/" />
  <updated>2011-12-13T18:30:02Z</updated>
  <author>
    <name>Jan Novák</name>
    <email>novak@example.com</email>
  </author>
  <id>urn:uuid:60a76c80-d399-11d9-b91C-0003939e0af6</id>
  <entry>
    <title>Článek zcela o ničem</title>
    <link href="http://example.org/clanek.html"/>
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
    <updated>2011-12-13T18:30:02Z</updated>
    <summary>Popis článku.</summary>
  </entry>
</feed>
```

Ještě ukázka začlenění *Atomu* do *XHTML* dokumentu pomocí elementu `<LINK>`:

```
<link href="atom.xml" type="application/atom+xml" rel="alternate"
title="Odkaz na čtečku Atom" />
```

⁶³ Verze 2.0 neměla být ke všemu vůbec kompatibilní s těmi předchozími.

⁶⁴ Že je *Atom* spíše paralela k *RSS* napovídá i použití stejného loga.

⁶⁵ Agregace dat na internetu je stahování obsahu z několika zdrojů na jedno místo, obvykle v pravidelných časových intervalech. Syndikace dat je publikování částí stránek (typicky aktualit, diskuzí, nových produktů apod.) na jiné stránce.

3.12 FBML

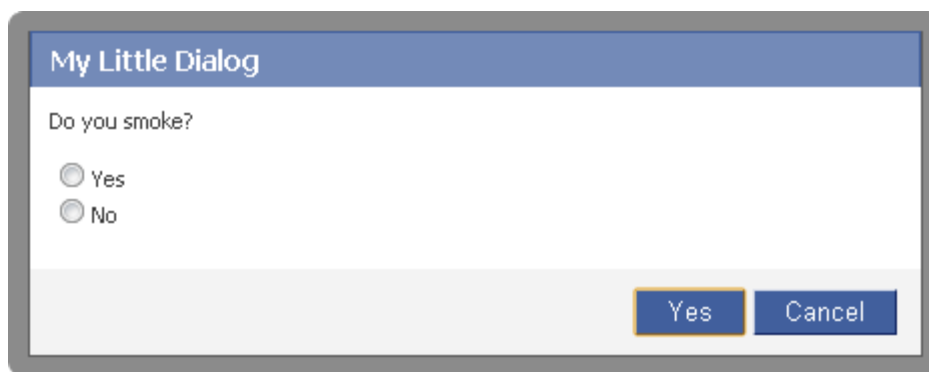
Sociální síť *Facebook* není třeba dlouze představovat. Ačkoliv vznikla teprve v roce 2004, je v současnosti nejrozšířenější aplikací tohoto typu na světě s více než 600 mil. uživatelských účtů.⁶⁶

Od roku 2007 je v každém profilu pomocí jazyka *FBML* možné upravovat jeho vzhled a funkčnost. *API* (používající také upravený *JavaScript* – *FBJs* a *DB* dotazovací jazyk *FQL*) zajišťuje přesměrování podle prohlížeče, odlišení stránek pro „přátele“ a ostatní uživatele, vkládání obrázků či jiných objektů, dialogová okna apod. *FBML* má vlastní stránku s informacemi pro vývojáře,⁶⁷ nicméně jeho rozvoj byl v březnu 2011 ukončen a je nahrazen platformou *iFrames*.

Syntaxe *FBML* je velmi podobná klasickému *HTML*, elementy však obsahují prefix *fb*. Následuje ukázka kódu pro dialogové okno:

```
<fb:dialog id="supa-dialog" cancel_button=1>
  <fb:dialog-title>My Little Dialog</fb:dialog-title>
  <fb:dialog-content>Do you smoke?
    <form id="supa-form">
      <p/>
      <input type="radio" name="supa-radio" value="Yes"/>Yes
      <br/>
      <input type="radio" name="supa-radio" value="No"/>No
    <p/>
    </form>
  </fb:dialog-content>
  </fb:dialog-button type="submit" value="Yes" form_id="supa-form"/>
</fb:dialog>
```

Výsledek na *Facebooku*:



Obr. 21: Dialogové okno ve *Facebooku* pomocí jazyka *FBML*
[získáno: 05-29-2011; zdroj: autor].

⁶⁶ Údaj k lednu 2011. Více informací zde:

http://www.msnbc.msn.com/id/40929239/ns/technology_and_science-tech_and_gadgets/

⁶⁷ Více informací zde: <http://developers.facebook.com/docs/reference/fbml/>

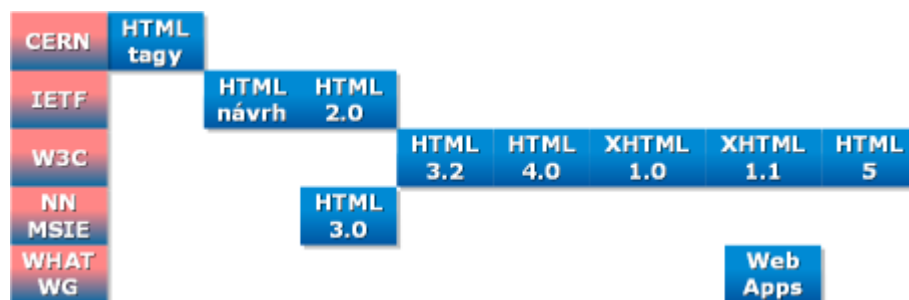
3.13 HTML5

3.13.1 Obecné vlastnosti

Několik posledních let na webu je ve znamení aplikací. Internetové stránky již dávno přestaly být jednoduchými výpisy textů a obrázků.⁶⁸ Několik posledních let zároveň ale chybí (komplexní) nástroj, který by poskytoval pevnou a jednotnou základnu pro bohatý multi-mediální obsah, přitom zachovával zpětnou kompatibilitu se stávajícími technologiemi, fungoval ve všech prohlížečích a na různorodých zařízeních, produkoval čistý a sémantický kód, mohl být zpracováván strojově, byl přijímán, podporován a používán všemi důležitými subjekty na poli webových technologií apod.

Jak zdůraznily předchozí kapitoly, jazyky *HTML* a *XHTML* léta stagnovaly. Webové technologie se skokově rozvíjely z roku na rok; bylo tedy třeba je složitými postupy integrovat do struktury hypertextových dokumentů staré bezmála dvě desetiletí. Novými fenomény se po roce 2000 staly výkonné mobilní telefony s přístupem na *WAP* a internet; v posledních dvou třech letech pak také tablety a čtečky elektronických knih s *Wi-Fi* adaptéry. Tato zařízení vyžadují optimalizaci pro své menší (často dotykové) displeje či podporu odlišných technologií. Ignorovat je znamená zřít se části „surfujících“ uživatelů, která roste o desítky procent ročně.

HTML5 samozřejmě nevzniklo najednou z ničeho nic a není to jen oživený pokračovatel čtvrté verze tohoto značkovacího jazyka. Je to *framework*, soubor technologií a postupů vycházejících z iniciativy sdružení *WHATWG*, které vzniklo v roce 2004 a vypracovalo projekty *WebApplications* a *WebForms*.



Obr. 22: Geneze (X)HTML v letech 1991 – 2011
[získáno: 05-05-2011; zdroj: autor].

V roce 2007 se tyto návrhy přesunuly pod správu konsorcia *W3C*. V té době se pracovnímu návrhu začalo říkat právě *HTML5*. Asi od roku 2010 některé stabilní části specifikace již vývojáři aplikují na nových stránkách,⁶⁹ vzhledem ke složitosti a komplexnosti celého projektu je však jeho finalizace zatím v nedohlednu.⁷⁰ To má zákonitě své odpůrce upozorňu-

⁶⁸ Začalo to jednoduchým *JavaScriptem* v pol. 90. let. O nějaký rok později přišel *Macromedia* (dnes *Adobe*) *Flash*, což je ovšem uzavřená komerční platforma. V roce 2005 přišel *AJAX*, nastala éra *frameworků*. *Microsoft* vydal *Silverlight* (další uzavřená platforma). Všem těmto technologiím má konkurovat *HTML5* a má je nahradit, příp. je využívat (*JS*, audio a video kodeky apod.).

⁶⁹ Označení stránka (*site*) či pohyb v historii webového prohlížeče, začínají být překonanými pojmy. Webové aplikace žádnou historií mít nemusí, běží celou dobu v jednom okně, i když je to několik dynamicky generovaných podstránek (*sub-sites*) dohromady. O historii nelze mluvit také v případě, kdy se jedná např. o přehrávač videí nebo *javaskriptový framework*, který nesáhá po dobu své činnosti na server. V souvislosti s těmito změnami se nabízí otázka, jakou budoucnost mají dříve v liště prohlížečů dominantní tlačítka Vpřed a Zpět.

⁷⁰ Odhady hovoří o tom, že kompletní specifikace včetně testování a oprav bude hotová v roce 2022. Více informací zde: <http://www.root.cz/zpravicky/kdy-bude-hotovo-html-5/>

jící na (typickou) pomalost práce konsorcia, neřkuli úplnou nadbytečnost *HTML5*, protože odpovídající (nicméně však komerční) technologie jsou na vyšší úrovni a k dispozici již dnes.⁷¹

Konsorcium představuje vlastnosti a schopnosti *HTML5* v osmi bodech:

- Sémantický kód: více logických značek, snadnější zpracování roboty a lidmi
- Podpora *offline* aplikací: lokální úložiště dat, synchronizace
- Přístup k hardwaru klienta: mikrofony, kamery apod.
- Lepší konektivita: podpora nových komunikačních protokolů
- Multimedia: přímá podpora audio a video formátů
- Grafické efekty: podpora *Canvas*,⁷² *CSS3*, *SVG*, *WebGL*⁷³
- Vyšší výkon a integrace: *all-in-one*, není potřeba pluginů a produktů třetích stran, což znamená vyšší rychlost a stabilitu aplikací



Obr. 23: Logo *HTML5*

[získáno: 05-05-2011; zdroj: Příspěvatelé Wikipedie. Dostupné z WWW: <http://cs.wikipedia.org/>].

3.13.2 Sémantika

Hned první dvě novinky přinášejí vývojářům okamžitou a velikou úlevu při tvorbě stránek. Jedná o prostou definici dokumentu jako *HTML* a jednoduché zvolení použitého kódování. Není potřeba již žádných definic *DTD* a složitých meta-tagů.

Následující 15ti znakový zápis zaručuje, že *HTML* dokument poznají všechny starší prohlížeče a vykreslí stránku ve standardním režimu. Nový meta-tag pod ním předepíše znakovou sadu. Zápis je bezpečně kompatibilní se staršími prohlížeči:

```
<!DOCTYPE html>
<meta charset="windows-1250">
```

Zmíněné dvě novinky lze okamžitě bez obav používat na všech stránkách. Především kvůli lepšímu strojovému zpracování logické struktury se v *HTML5* zavádí velké množství nových značek. Některé starší či dříve zavržené dostávají pozměněný význam. Nové elementy mají pochopitelná jména a snahou tvůrců bylo, aby postupně nahradily univerzální nevýznamový element `<div>`. Např. místo zápisu:

```
<div id="header"></div>
```

Je zaveden odpovídající element:

```
<header></header>
```

⁷¹ Kritický článek na téma *HTML5* napsal např. R. Hulán. Více informací zde:

<http://myego.cz/item/nadseni-z-html5-neni-na-miste>

⁷² *Canvas* je „malířské plátno“, nový *HTML* tag, do kterého lze vykreslovat dvourozměrnou a bitmapovou grafiku. Je to novější paralela k jazyku *SVG*, programuje se pomocí *JavaScriptu*.

⁷³ Nový projekt vykreslování 3D scén ve webovém prohlížeči. Ideový nástupce *VRML* a *X3D*. Opět využívá *Canvas* a jako jazyk *JavaScript*.

Výsledkem je sémanticky hodnotnější kód, navíc je kratší a přehlednější.⁷⁴ Typická stránka v elementu `<body>` pak bude vypadat třeba nějak takto:

```
<body>
  <header>
    <nav>
      <ul>
        <li>Položka menu</li>
      </ul>
    </nav>
  </header>
  <section>
    <article>
      <header>
        <h2>Nadpis článku</h2>
        <p>Publikoval<time datetime="2011-05-09T12:00:00+01:00">
          9. května 2011</time> <a href="#">Ondřej Smíšek</a></p>
      </header>
      <p>Text článku.</p>
    </article>
  </section>
  <footer>
    <p>Copyright &copy; 2011 Ondřej Smíšek</p>
  </footer>
</body>
```

3.13.3 Grafika, video a zvuk

Při návrhu prvního *HTML* nikdo s audio a video soubory nepočítal. Zpočátku představovaly jisté ozvláštnění animované a/nebo průhledné obrázky ve formátu *GIF*. Microsoft také přišel s proprietárními elementy `<BGSOUND>`, `<MARQUEE>` apod. Po vzniku `<EMBED>`, kterýžto představil *Netscape* pro svůj *Navigator*, již bylo možné vkládat do stránek produkty třetích stran, jako *Java*, *Media Player*, *QuickTime* či dnes nejrozšířenější *Flash* přehrávače. Později přišel `<OBJECT>`, nicméně k přehlednému a sémantickému kódu to mělo velmi daleko, nehledě na potřebu neustálé aktualizace. S boomem video serverů se stránky stávaly velmi pomalé a indexovací stroje si s nimi nevěděly rady.

HTML5 zavádí jednoduché elementy `<audio>` a `<video>`. Vložení videa do stránek může vypadat takto:

```
<video src="./video_soubor.ogv"
  width=320
  height=240
  controls
  poster="./obrazek_při_nacitani.jpg">
  Váš prohlížeč nepodporuje element „video“, soubor si můžete
  stáhnout <a href="./video_soubor.ogv">zde</a>
</video>
```

Vzhled přehrávače lze samozřejmě ovlivňovat javaskriptem a kaskádovými styly. Jediným problémem dosud zůstává nejednoznačná implementace kodeků. *Firefox* a *Opera* podporují otevřený formát *OGG*, *Apple* produkty *MP4*, *Chrome WebM*. S novými verzemi prohlížečů bude již situace jasnější (např. *Firefox 4* implementuje *WebM*), jako první jde z kola (zřejmě kvůli komerčním tlakům) volný *OGG*.

Obdobný zápis je použit i u elementu pro přehrávání zvuků či písní. Zde soupeří formáty *MP3*, *Ogg Vorbis* a *WAV*:

⁷⁴ *Internet Explorer* elementy, které nezná, neumí stylovat. Řeší se to pomocí *JavaScriptu* nebo je možné také s úspěchem použít mikroformátů.

```
<audio controls preload>
  <source src="./zvuk.ogg">
  <source src="./zvuk.mp3">
</audio>
```

Grafickou podobu stránek již od roku 1996 ovlivňují kaskádové styly. *HTML5* nyní přichází s „plátnem“ (*canvas*), což je oblast, definovaná v samotném *HTML* šířkou a výškou, ve které lze dynamicky vykreslovat grafiku jaskriptem. Dostupné funkce jsou podobné grafickým programům. Využití elementu `<canvas>` je bohaté – grafy, animace, hry, úpravy obrázků apod. Podpora v prohlížečích je již dostatečná.

Ukázka začlenění do *HTML* kódu:

```
<canvas id="platno" width="200" height="200">
  Pokud vidíte tento text, Váš prohlížeč nepodporuje Canvas.
</canvas>
```

A samotný skript, který vykreslí v okně prohlížeče modrý čtverec:

```
var example = document.getElementById('platno');
var context = example.getContext('2d');
context.fillStyle = "rgb(0,0,255)";
context.fillRect(25, 25, 50, 50);
```

3.13.4 Webové aplikace

Aplikace, využívající internetový prohlížeč pro svůj běh, jsou už několik let na denním pořádku. Jejich možnosti ale dosud limitovaly jak samotné browsery, tak nestavový protokol *HTTP*.

Moderní prohlížeče zpracovávají velmi rychle *JavaScript*, který je nejčastějším „pohonem“ webových aplikací. Spolu s *HTML5* se objevil koncept *WebWorkers*, což jednoduše znamená schopnost několika vláknového zpracovávání skriptů. Pomocné výpočty mohou běžet paralelně s hlavním tokem programu. To urychlí běh aplikace při komplikovanějších úlohách.

HTML5 přidává dále možnost *offline* práce, protože umožňuje ukládat pracovní data na klientském počítači⁷⁵. Webové služby díky tomu mohou pracovat i bez stálého připojení, což je výhodné např. u přenosných zařízení. Po obnovení připojení je program schopen synchronizace. Technologie se nazývají *AppCache* a *WebStorage*.

Má-li aplikace fungovat v *offline* módu, je nutné vytvořit tzv. *cache manifest*, což je seznam souborů, jež mají být přístupné lokálně. Jméno souboru se uvede jako atribut kořenového elementu `<html>`:

```
<html manifest="./cache.manifest">
```

3.13.5 Formuláře

Formuláře pro zadávání dat uživatelem se od vzniku specifikace *HTML 2.0* nijak zvlášť nezměnily. K jejich ovládání se používají klientské a serverové skripty, vzhled lze ovlivnit kaskádovými předpisy. Co do funkčnosti, nové *HTML5* zavádí sofistikovanější typy políček a pro všechna ostatní také nové možnosti. Zpětná kompatibilita je zaručena, tudíž lze vylepšené možnosti webových formulářů používat již dnes.

Všechna nová tlačítka začínají obvyklou formulí `<input type=...,` přičemž k dispozici jsou následující možnosti:

⁷⁵ Koncept je převzat z produktu *Google Gears*, který už byl zrušen.

- `datetime` – datum a čas
- `date` – datum
- `month` – měsíc
- `week` – týden
- `time` – čas
- `number` – číslo
- `range` – posuvník s rozsahem číselných hodnot
- `email` – e-mailová adresa včetně ověření správnosti formátu
- `url` – URL adresa
- `search` – vyhledávací políčko
- `color` – výběr barvy a její převedení do textového formátu

Je zřejmé, že přednastavené hodnoty a vzhled budou závislé na zařízení, tudíž např. u dotykových tabletů lze očekávat u políčka `number` přímo číselnou klávesnici apod. Pokud prohlížeč specifické políčko nezná, použije se výchozí typ `text`, tudíž funkčnost lze zaručit konvenčním způsobem. Zcela se vyhnout skriptování na straně klienta lze díky volitelnému příznaku povinného vyplnění políčka či omezení zadávaných hodnot. Doposud šlo přitom pouze omezovat rozsah znaků v poli.

3.13.6 Další vlastnosti

V tomto posledním oddíle budou rychle představeny tři zajímavé možnosti *HTML5*. Jedná se o schopnost práce s lokálními soubory klienta, implementace metody *drag-and-drop* a možnost definice vlastních datových atributů k elementům.

K existujícímu *DOM* bylo přidáno také souborové *API*, s jehož pomocí lze jednoduše provádět některé operace se soubory, které bylo až dosud nutno řešit pluginy a javascriptem. Např. tak lze přímo metodou *drag-and-drop* přetahovat přílohy do okna webového poštovního klienta.⁷⁶ Aplikace běžící v prohlížeči se tím významně přiblížily standardu desktopového rozhraní. Ostatně fy *Google* vidí budoucnost jasně, když nedávno představila svůj ultra-přenosný laptop, který neobsahuje operační systém – protože si vystačí jen s vylepšeným prohlížečem *Chrome*.⁷⁷ Prodej laptopu byl zahájen v červnu 2011 a *Google* se nyní bude snažit přesvědčit uživatele, že v *online* kyberprostoru mohou mít všechna data, dělat zde všechnu běžnou práci a bavit se.

Pro potřeby implementace *drag-and-drop GUI* není v *HTML5* třeba spoléhat se na objemné externí *frameworky*. Vše zařídí nová specifikace a *JavaScript*. Stačí mít dva elementy, např. `<div>`, přičemž jeden bude ten přesouvaný a druhý kontejner na jeho umístění.

Co se týče vkládání datových atributů do elementů, není to tak mocný nástroj, jako syntaxe *XML* s vlastními značkami. Inspirace však nepochybně vzešla odsud a možnosti této novinky jsou tak bohaté, jak dovolí nápaditost vývojářů. Jméno atributu musí začínat prefixem `data-`, hodnotou může být libovolný znakový řetězec.

Např. takovýto běžný seznam:

```
<ul id="motory">
  <li data-kde="Camaro, Corvette">Chevrolet V8 305cid.</li>
  <li data-kde="Electra, Skylark">Buick V8 350cid.</li>
</ul>
```

V *HTML5* bude doplněn o funkci, kdy po najetí na položku *JavaScript* zajistí zobrazení tabulky s typy vozidel, ve kterých byl konkrétní motor použit, a to bez nutnosti použití *frameworků*. Datové atributy totiž *JavaScript* automaticky zahrne do *DOM* stránky.

⁷⁶ Testováno na službě *GMail* fy *Google*.

⁷⁷ Více informací zde: <http://www.google.com/chromeos/features.html>

Kapitola snad dostatečně objasnila široké možnosti nového *HTML* standardu. Nezbyvá než pevně doufat v rychlou implementaci všech vlastností do moderních prohlížečů. Na webu by tím mělo dojít k všestrannému zlepšení služeb, ať už se jedná o sémantiku, jednodušší vývoj či příjemné grafické prostředí, které bude za všech okolností vypadat stejně a bude funkční.

Následující ukázka (obr. 23) představuje stránku napsanou v *HTML5*. Zdrojový kód je skutečně otevřený, o čemž se může každý návštěvník snadno přesvědčit po příslušné volbě ve webovém prohlížeči. Na interaktivní stránce je simulace otáčení listů, obrázek animovaný, text ve dvou sloupcích apod., přitom lze však veškerý text a grafiku označovat a běžně s nimi manipulovat. To je síla *HTML5* – bezproblémové oddělení syntaxe, vzhledu a funkčnosti, animace a zvuk bez produktů třetích stran, nižší hardwarové nároky, možnosti přizpůsobení konkrétním uživatelům a zobrazovacím prostředkům, pochopitelná metadata pro indexování roboty.



Obr. 24: *HTML5* stránka

[získáno: 07-11-2011; zdroj: <http://www.20thingsilearned.com/cs-CZ/home>].

4 Závěr

Předložená bakalářská práce sledovala vývoj značkovacích jazyků napříč půlstoletím jejich existence. Tak jako všechny technické prostředky i značkovací jazyky vznikly za účelem zjednodušení a zpřehlednění práce.

Postupně jejich princip proniká všude tam, kde je nasazena výpočetní technika. Během desetiletí se objevilo několik oblastí využití značkování, přičemž nové technologie a postupy samozřejmě ovlivňovaly ty předchozí.

Tyto oblasti jsou:

- *Typografie*. Značkování v typografických programech řídí vzhled dokumentu, podílí se na tvorbě automatických textů, jako seznamů, poznámek pod čarou, citací, obsahů či rejstříků. Umožňuje rychlou tvorbu profesionálních dokumentů jednotného vzhledu, případně jednoduchou aplikaci změn.
- *Dokumentace*. Dokumentační programy vznikly na základě typografických potřeb a mohutného rozvoje spotřební techniky a elektroniky, čemuž ovšem předcházely vojenský a vesmírný výzkum.⁷⁸ Nabízejí obvyklé funkce textového procesoru, dále sledování verzí a změn v nich, a především velké množství různorodých výstupů na základě jednoho datového vstupu – pro tisk, na webovou stránku s hypertextovou navigací, uzavřený či *XML-like* elektronický dokument, formát pro mobilní telefon, tablet apod. Dokumentační formáty se velmi hodí také pro interaktivní nápovědy k programům.⁷⁹
- *Hypertextové systémy*. Hypertext byl na akademické půdě znám několik desetiletí, na praktickou aplikaci však čekal až do 80. let, kdy nastoupily osobní počítače. Snahou vývojářů a později standardizačních organizací bylo oddělení prezentační a sémantické složky hypertextových dokumentů. V posledních dvaceti letech pak ta samá situace nastala u webových stránek tvořených v jazycích *HTML*, *XHTML*, nebo jejich nesčetných odnožích. Problematika významu a použití značek není zcela černobílá – neustále se hledají přechodná řešení kvůli kompatibilitě s prohlížeči a dalším softwarem, zažitým postupům i tlakům vývojářských společností. S pokračujícím vývojem a implementací *HTML5* bude tento problém zase o něco menší; je ovšem jisté, že budou nadále koexistovat další alternativní řešení. Konečně je dobré říci, že většina běžných uživatelů si pod pojmem značkovací jazyk představí právě nějakého zástupce pro zápis internetových stránek.⁸⁰
- *Metadata*. Informace, které významově popisují samotná data a zpřístupňují je k hlubší automatizované analýze či zařazení, se nazývají metadata. Ať už se v aplikačním rozhraní používá jakýkoliv metadatový formát, s největší pravděpodobností bude vycházet přímo – nebo alespoň z principů – *SGML* a *XML*. Bez metadat by byl kyberprostor neprohlédnutelnou a neprohledatelnou pavučinou, ve které se všechno ztrácí. Díky značkování jsou možné dnes samozřejmé procesy jako indexace, ukládání a vyhledávání.

⁷⁸ Americká *NASA* používala pro dokumentaci k programu *Apollo* jeden z prvních použitelných značkovacích systémů.

⁷⁹ Např. nápověda *OS Windows*, obsahující mnoho úrovní vnoření a dalších odkazů je s výhodou založena na *HTML*.

⁸⁰ Také autor této práce, než hlouběji pronikl do problematiky, měl velmi mlhavou představu o šíři značkovacích systémů.

- *Skriptování a programování.* Na značkování jsou založeny také některé skriptovací a programovací jazyky pro popis virtuálního prostoru, bitmapové a vektorové grafiky, tvorbu aplikačních rozhraní aj. Tyto produkty se na rozdíl od „běžných“ programovacích postupů vyznačují větší pochopitelností pro méně matematicky založené osoby, to ale neznamená, že nedosahují plnohodnotných výsledků. Dalším specifickým je, že nemají vlastní překladač (*compiler*), ale jejich frameworkem je typicky webový prohlížeč. V době všudypřítomného, rychlého, levného a spolehlivého internetového připojení již nelze tuto skutečnost považovat za nevýhodu. Dnešní webové frameworky stojí na precizní spolupráci mezi značkovacími, prezentačními, dotazovacími a skriptovacími jazyky. Webové prohlížeče jsou zdarma a přitom nabízejí obrovskou podporu pro různorodé platformy. Rychlý vývoj a implementace nových technologií v rámci konkurenčního boje a prestiže vývojářských skupin je ku prospěchu každého uživatele, kodéra či programátora. Zpětná kompatibilita je správnou cestou k tomu, aby se koloběh „vynalézání kola“ zastavil a bez obav se stavělo na prověřených základech.
- *Syndikace a agregace dat.* Ideální využití pro značkové jazyky jsou syndikační a agregační formáty internetových stránek. Vytvářejí se tím na webu místa, která koncentrují vybrané informace z libovolného množství dalších zdrojů. Tyto údaje se navíc mění tak, jak se tomu děje přímo u zdroje. Personalizací syndikačních a agregačních zdrojů dostává uživatel mocný informační nástroj.

V minulých letech se zbytečně mnoho úsilí věnovalo některým formátům, které se neujaly. To zčásti zapříčinil přirozený vývoj, zčásti však cílený tlak a snaha o zviditelnění některých komerčních společností a organizací. Příliš mnoho podobných technologií napáchalo víc škody než užitku – nové lidi mátklo a odborníkům přidělovalo práci.

Budoucí trend je nastolen dobře. Na poli syndikace dat by měl dominovat praktický *Atom*. Pro webovou grafiku se výtečně hodí *SVG*. Na metadata neexistuje nic lepšího než *XML*. Pro *WWW* stránky pak jednoznačně *HTML5*. V oblasti *DTP* není důvod k opuštění *TeXu*, *PostScript* bude zřejmě definitivně nahrazen *PDF*. Bezpochyby také *RTF* a *DocBook* mají v příštích letech co nabídnout.

Značkové systémy přinášejí do současného světa informačního přetížení řád. Nelze očekávat, že bude situace ohledně tohoto přehlcení lepší – právě naopak. Značkování je jedním ze způsobů, jak lidstvo může efektivně pořádat, vyhledávat, a také prezentovat své znalosti. Díky smysluplnému třídění nebude problém předávat soudobé informace dalším generacím.

Slovník zkratek

Slovník neobsahuje notoricky známé a zavedené zkratky (*CD-ROM*, *ICT* apod.). Pojmy, které nejsou uvedeny, s největší pravděpodobností nejsou zkratkami a jejich vysvětlení lze nalézt přímo v místě výskytu (typicky v poznámce pod čarou). Zkratky, jejichž význam je předmětem samotného textu, nejsou již více rozváděny (*HTML* apod.).

AJAX: *Asynchronous JavaScript and XML*. Soubor technologií (*(X)HTML*, *CSS*, *DOM*, *JavaScript*, *XHR* apod.) pro vývoj interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti obnovování uživatelem.

ANSI: *American Normalization Standard Institute*. Nezisková instituce pro technickou normalizaci v USA, založena v roce 1918.

API: *Application Programming Interface*. Rozhraní pro programování aplikací.

ARPA: *Advanced Research Projects Agency*. Agentura pro výzkum pokročilých obranných projektů při ministerstvu obrany USA, založena v roce 1958 na základě vypuštění sovětské družice Sputnik 1. Jejím cílem je udržovat vojenský náskok USA. Stojí např. za projektem *ARPANET*. Od roku 1972 nese název *DARPA* (*Defense Advanced Research Projects Agency*).

ARPANET: *Advanced Research Projects Agency Network*.

ASCII: *American Standard Code for Information Interchange*. Americký standardní kód pro výměnu informací, definuje znaky používané v informatice. První verze dokončena v roce 1963.

AVI: *Audio Video Interleave*. Multimediální kontejner fy *Microsoft* uvedený v roce 1992. Je zastaralý, ale stále hojně používaný a podporovaný.

BBS: *Bulletin Board System*. Elektronická diskusní fóra v předwebovém internetu.

BITNET: *Because It's Time NETwork*. Americká akademická síť, fungující v letech 1981-2007, která svého času propojovala kolem 500 institucí.

BMP: *BitMaP*. Grafický formát pro rastrovou grafiku.

CAD: *Computer-Aided Design*. Činnost projektování pomocí *IT*.

CAPTCHA: *Completely Automated Public Turing test to tell Computers and Humans Apart*. Plně automatický veřejný Turingův test k odlišení počítačů a lidí; používá se na *WWW* tam, kde je potřeba jednoduše rozlišit lidské uživatele od robotů, např. při vkládání komentářů, při registraci apod.

CDF: *Channel Definition Format*. Formát webové agregace dat fy *Microsoft*. Předchůdce *RSS*. Nativní podpora v prohlížeči *Internet Explorer* již skončila.

CERN: *Conseil Européen pour la Recherche Nucléaire*. Evropská organizace pro jaderný výzkum. V roce 1989 na její půdě vznikl web.

CESNET: Sdružení založené roku 1996 českými *VŠ* a *AV*. Stará se o rozvoj a provoz celorepublikové akademické počítačové sítě.

CGI: *Common Gateway Interface*. Serverový skriptovací jazyk.

C-HTML: *Compact-HyperText Markup Language*.

CLI: *Command Line Interface*. Komunikace s programem či *OS* probíhá postupným zadáváním příkazů uživatelem.

CSS: *Cascading Style Sheets*. Kaskádové stylové předpisy pro definici vzhledu *HTML* dokumentů. Podřazené elementy typicky dědí vlastnosti svých rodičů, takže se nemusí znovu definovat.

CSTUG: *CzechoSlovakian TeX Users Group*. Lokální sdružení uživatelů *TeXu*.

DC: *Dublin Core*. Metadatový formát fy *OCLC*.

DEC: *Digital Equipment Corporation*. Již zaniklá americká počítačová firma. Sloučena s fy *Compaq*, kterou dále koupilo *HP*.

DHTML: *Dynamic HyperText Markup Language*. Koncept fy *Microsoft*; spojení technologií *DOM* – *HTML* – *CSS* – *JavaScript/JS*cript/*VBScript* pro tzv. dynamické *WWW*.

DNS: *Domain Name System*. Zabezpečuje převod jmenných názvů domén na *IP* adresy a mnohé další síťové služby.

DOI: *Digital Object Identifier*. Jedinečný identifikátor digitální entity.

DOM: *Document Object Model*. Objektový model dokumentu, každá jeho část je oddělená a samostatně přístupná k úpravám.

DOS: *Disc Operating System*.

DSSSL: *Document Style Semantics and Specification Language*.

DTD: *Document Type Definition*. Jazyk pro popis struktury *SGML* a *XML* dokumentů a jejich odvozenin.

DTP: *DeskTop Publishing*. Počítačová příprava tiskových dokumentů.

DXF: *Drawing Exchange Format*. CAD otevřený (výměnný) formát fy *Autodesk*.

EARN: *European Academic Research Network*. Evropská akademická počítačová síť, svého času také brána k datovému propojení Evropy a USA.

ECMA: *European Computer Manufacturers Association*. Nezisková standardizační organizace evropských počítačových firem, založena v roce 1961.

FBJS: *FaceBook JavaScript*.

FBML: *FaceBook Markup Language*.

FQL: *Facebook Query Language*.

FTP: *File Transfer Protocol*. Internetový protokol pro přenos textových či binárních souborů.

GIF: *Graphics Interchange Format*. Grafický formát z roku 1987 o barevné hloubce 8 bitů, používající bez-ztrátovou kompresi a umožňující animace.

GML: *Generalized Markup Language*. Zobecněný značkovací jazyk fy *IBM*.

GNU: *Gnu's Not Unix*. Projekt otevřeného OS unixového typu.

GPO: *Government Printing Office*.

GSM: *Groupe Spécial Mobile*. Globální systém pro mobilní komunikaci, nejrozšířenější digitální standard pro mobilní telefony.

GUI: *Graphical User Interface*. Grafické uživatelské rozhraní, které používá interaktivní grafické ovládací prvky pro interakci s uživatelem. První GUI navrhli v roce 1973 v *Xerox PARC*.

HCI: *Human-Computer Interaction*. Vědní oblast zabývající se problematikou komunikace člověka s počítačem.

HDML: *Handheld Device Markup Language*. Značkovací jazyk pro přenosná zařízení. Záhy nahrazen *WML* a *XHTML Mobile*.

HTML: *HyperText Markup Language*. Hypertextový značkovací jazyk.

HTTP: *HyperText Transfer Protocol*. Internetový protokol pro výměnu hypertextových dokumentů.

HTTPS: *HyperText Transfer Protocol Secured*. Zabezpečený protokol *HTTP*.

IBM: *International Business Machines*. Jedna z největších technologických firem na světě, založena v roce 1888.

IANA: *Internet Assigned Numbers Authority*. Organizace dohlížející na přidělování *IP* adres, správu primárních domén a internetové protokoly.

ICANN: *Internet Corporation for Assigned Names and Numbers*. Nezisková organizace dohlížející na internetové technologie, založena v roce 1998.

IEC: *International Electrotechnical Commission*. Nezisková organizace pro elektrotechnickou standardizaci, založena v roce 1906.

IETF: *Internet Engineering Task Force*. Další z organizací vyvíjejí internetové standardy (hlavně protokoly), spolupracuje s *W3C*.

IPN: *Internet Protocol Number*. Číslo jednoznačně identifikující rozhraní (např. *Ethernet* kartu) v síti. Rozsah čísel je 2^{32} v protokolu *Ipv4*, který je zatím nejrozšířenější. Vzhledem k nedostačujícímu počtu adres (protokol je z roku 1981) se v současnosti přechází na 2^{128} rozsah v protokolu *IPv6*.

ISO: *International Organization for Standardization*. Celosvětová organizace sdružující národní standardizační úřady, založena v roce 1947.

ISOC: *Internet Society*. Mezinárodní nezisková organizace koordinující vývoj internetových standardů. Založena v roce 1992.

ISO/OSI: *ISO/Open System Interconnection*. Standard počítačové sítě z roku 1984. Pracuje se sedmi vrstvami, od fyzické po aplikační. Ujal se ale levnější (a horší) standard *Ethernet*.

ISP: *Internet Service Provider*. Poskytovatel internetového připojení.

IT: *Information Technologies*.

JPEG: *Joint Photographic Experts Group*. Název konsorcia a zároveň grafického formátu používajícího ztrátovou kompresi dat.

JSSS: *JavaScript Style Sheets*. Stylový jazyk pro webové dokumenty fy *Netscape* z roku 1996. Převálcován CSS a zapomenut.

LAMP: *Linux – Apache – MySQL – PHP/Perl/Python*. Soubor *open-source* technologií, které se nejčastěji používají při vývoji webových aplikací.

LAN: *Local Area Network*. Lokální propojení počítačů.

MIDI: *Musical Instrument Digital Interface*. Standard z roku 1983 pro komunikaci mezi digitálními hudebními nástroji.

MIME: *Multipurpose Internet Mail Extensions*. Internetový standard pro e-mailovou komunikaci zavádějící podporu národních znakových sad, příloh, vícedílných zpráv apod.

MIT: *Massachusetts Institute of Technology*. Významná univerzita v USA.

MPEG: *Motion Picture Experts Group*. Pracovní skupina a standard pro kódování audiovizuálních dat používající ztrátovou kompresi.

NASA: *National Aeronautics and Space Administration*. Národní úřad pro letectví a kosmonautiku USA. Založena v roce 1958 po sovětském úspěchu s první umělou družicí Země.

NCP: *Network Control Program*. Původní transportní protokol *ARPANETu*.

NCSA: *National Center for Supercomputing Applications*. Technologický ústav v Illinois, USA, založen v roce 1986.

NSF: *National Science Foundation*. Agentura na podporu vědy v USA.

NSFNET: *National Science Foundation NETwork*. Páteční síť internetu v USA v letech 1985-1995.

OASIS: *Advancing Open Standards for the Information Society*.

OCLC: *Online Computer Library Center*. Firma zabývající se automatizací a organizací knihoven v celosvětovém měřítku, tvoří např. *OPAC WorldCat*. Založena v roce 1967.

OGG: Svobodný multimediální formát.

OPAC: *Online Public Access Catalogue*. Elektronický veřejný knihovní katalog, první vznikl v roce 1975. V současné době *OPACy* integrují služby webu 2.0.

OS: *Operating System*. Základní ovládací software počítače sloužící k *HCI*.

OS/2: *Operating System/2*. OS fy *IBM* z roku 1987.

OWL: *Web Ontology Language*. Jeden z jazyků sémantického webu 2.0.

PARC: *Palo Alto Research Facility*. Vývojové oddělení fy *Xerox*, které dalo vzniknout technologiím jako *GUI*, *Ethernet*, počítačová myš atd.

PC: *Personal Computer*. Průmyslový standard zahrnující širokou škálu norem a doporučení pro domácí počítač.

PCL: *Printer Command Language*. Jazyk pro tiskárny společnosti *Hewlett-Packard*.

PDF: *Portable Document Format*. Formát pro ukládání dokumentů nezávislý na softwaru i hardwaru, nástupce *PostScriptu*. Oba formáty vyvinula firma *Adobe Systems*.

PDL: *Page Description Language*. Jazyk k popisu stránek.

PGML: *Precision Graphics Markup Language*. Značkovací jazyk pro vektorovou grafiku, nahrazen *SVG*.

PHP: *Hypertext PreProcessor*.

PNG: *Portable Network Graphic*. Grafický formát z roku 1996 o barevné hloubce 24 bitů, umožňuje průhlednost. Nástupce formátu *GIF*.

POP: *Post Office Protocol*. Internetový protokol ke stahování elektronické pošty ze serveru.

PPP: *Point-to-Point Protocol*. Protokol linkové vrstvy pro přímé propojení dvou síťových uzlů.

RDF: *Resource Description Framework*. Systém popisu zdrojů, internetový metadatový model.

RDP: *Remote Desktop Protocol*. Síťový protokol umožňující ovládat vzdálený počítač pomocí desktopového *GUI*.

RFC: *Request For Comments*. Volně dostupné dokumenty týkající se rozvoje internetových technologií.

RSS: *Really Simple Syndication*. Značkovací formát k syndikaci internetových zdrojů.

RTF: *Rich Text Format*. Formát souborů pro ukládání textů včetně formátovacích značek fy *Microsoft*.

SGML: *Standard Generalized Markup Language*. (Normovaný) univerzální značkovací jazyk.

SMIL: *Synchronized Multimedia Integration Language*. Jazyk pro tvorbu multimediálních prezentací. Jedná se o aplikaci *XML* od *W3C*.

SMTP: *Simple Mail Transfer Protocol*. Internetový protokol pro přenos elektronické pošty.

SQL: *Structured Query Language*. Dotazovací jazyk pro relační databáze. Standard z roku 1986.

SSH: *Secure Shell*. Zabezpečený komunikační protokol, nástupce *Telnetu*.

SVG: *Scalable Vector Graphic*. Značkovací jazyk pro vektorovou grafiku.

TCP/IP: *Transmission Control Protocol/Internet Protocol*. Sada protokolů pro komunikaci v počítačové síti internet.

Telnet: *Telecommunication protocol*. Internetový protokol pro připojení ke vzdálenému počítači.

TUI: *Text User Interface*. Rozhraní *OS* a programů používající textovou matici, ale na rozdíl od *CLI* pracují-cí s okny a myší.

TUG: *TeX Users Group*. Organizace sdružující uživatele *TeXu*.

UDP: *User Datagram Protocol*. Protokol nezaručující doručení dat v síti.

URL/URI: *Uniform Resource Locator/Uniform Resource Identifier*. Jednotný lokátor zdrojů, řetězec znaků s definovanou strukturou, který slouží k přesné specifikaci umístění dokumentů nebo aplikací na internetu.

USENET: Služba podobná *BBS*, ale bez centrálního serveru a řízení. Stejně jako *BBS* byl *USENET* překonán službou *WWW*.

VML: *Vector Markup Language*. Značkovací jazyk pro vektorovou grafiku, alternativa k *SVG*.

VRML: *Virtual Reality Markup Language*, později *Virtual Reality Modeling Language*. Jazyk pro popis prostorových scén.

W3C: *World Wide Web Consortium*. Mezinárodní společenství pro vývoj internetových standardů a doporučení. Založeno v roce 1994.

WAI: *Web Accessibility Initiative*. Iniciativa pro přístupný (bezbariérový) web.

WAN: *Wide Area Network*. Geograficky rozlehlá počítačová síť.

WAP: *Wireless Application Protocol*. Protokol mobilních elektronických služeb.

WAV: *WAVE form audio file format*. Zvukový formát fy *IBM* a *Microsoft* pro *PC*.

WBMP: *Wireless Bitmap*. Grafický 1bitový *BMP* formát pro mobilní telefony.

WCAG: *Web Content Accessibility Guidelines*. Pravidla přístupného *WWW* pro handicapované osoby. Za jejich vytvořením stojí skupina *WAI* v rámci *W3C*.

WHATWG: *The Web Hypertext Application Technology Working Group*. Sdružení, které vzniklo na základě nespokojenosti komerčních firem se standardizačními postupy *W3C*. V současnosti obě instituce spolupracují na specifikaci *HTML5*.

Wi-Fi: *Wireless Fidelity*. Soubor technologií a standardů pro bezdrátové datové sítě.

WIMP: *Windows, Icons, Menus, Pointing devices*. Model *GUI* zahrnující aplikační okna, jejich ikony, rozvíjející se menu (volby) a ovládání pomocí ukazovacího zařízení (myši, touchpadu apod.).

WLAN: *Wireless LAN*. Bezdrátová lokální síť, např. v domě či kanceláři.

WML: *Wireless Markup Language*. Jednoduchý značkovací jazyk pro mobilní telefony.

WWW: *World Wide Web*. Nejrozšířenější internetová služba současnosti.

WYSIWYG: *What You See Is What You Get*. Co vidíš, to dostaneš; toto označení se používá pro vývojové nástroje, které již v návrhovém rozhraní zobrazují dokument tak, jak se zobrazí na výstupním zařízení.

X3D: *eXtensible 3D*. *XML* jazyk pro popis a chování prostorových scén, nástupce *VRML*.

XDHTML: *eXtended Dynamic HyperText Markup Language*. Viz *DHTML*.

XHR: *XMLHttpRequest*. Komunikační rozhraní pro webové aplikace.

XHTML: *eXtensible HyperText Markup Language*. Rozšiřitelný hypertextový značkovací jazyk, slepá větev vývoje *HTML* se striktní *XML* syntaxí.

XML: *Extensible Markup Language*. Rozšiřitelný značkovací jazyk, nástupce *SGML*.

XSL: *eXtensible Stylesheet Language*. Rodina jazyků (*XML Transformations*, *XML Formatting Objects*, *XML Path*) pro prezentaci *XML* dokumentů.

Bibliografie

Tištěné publikace v češtině

1. BALVÍNOVÁ, A.; BÍLÝ, M. *Textové informační systémy : sázecí systém LaTeX : Cvičení*. 1. vyd. Praha : ČVUT, 1994. 64 s. ISBN 80-01-01201-8.
2. BRÁZDA, J. *XML : Praktické příklady : podrobný průvodce začínajícího uživatele*. 1. vyd. Praha : Grada, 2003. 212 s. ISBN 80-247-0699-7.
3. CASTRO, E. *XML pro World Wide Web : praktická vizuální příručka*. Brno : SoftPress, 2001. ISBN 80-86497-07-0.
4. CEJPEK, J. *Informace, komunikace a myšlení*. 2. přeprac. vyd. Praha : Karolinum, 2005. 234 s. ISBN 80-246-1037-X.
5. ČESENĚK, P. *Modemy, faxy, BBS a počítačová komunikace*. Vyd. 1. Praha : Grada, 1995. 184 s. ISBN 80-7169-171-2.
6. DELLWIG, I. *HTML 4 : příručka tvůrce webu*. 1. vyd. Praha : Grada, 2002. 272 s. ISBN 80-247-0297-5.
7. GERGELITSOVÁ, Š. *VRML v příkladech*. 1. vyd. Praha : BEN, 2004. 224 s. ISBN 80-7300-138-1.
8. HAVLÍČEK, Z. *Značkovací jazyky*. Praha, 2008. 41 s. Publikace k přednášce. Česká zemědělská univerzita, Provozně ekonomická fakulta, Katedra informačních technologií.
9. HEROUT, P. *XSLT 2.0 a SVG prakticky*. 1. vyd. České Budějovice : KOPP, 2010. 296 s. ISBN 978-80-7232-406-4.
10. HLAVENKA, J... [et al.]. *Vytváříme WWW stránky*. 6. akt. vyd. Praha : Computer Press, 2002. 356 s. ISBN 80-7226-748-5.
11. JAMSA, K... [et al.]. *Programování na Webu : průvodce vývojáře*. 1. vyd. Brno : UNIS Publishing, 1996. 664 s. ISBN 80-86097-03-X.
12. KOSEK, J. *HTML : tvorba dokonalých www stránek : podrobný průvodce*. Praha : Grada, 1998. 291 s. ISBN 80-7169-608-0.
13. KOSEK, J. *PHP : Tvorba interaktivních internetových aplikací*. Vyd. 1. Praha : Grada, 1999. 492 s. ISBN 80-7169-373-1.
14. KOSEK, J. *XML pro každého : podrobný průvodce*. 1. vyd. Praha : Grada, 2000. 164 s. ISBN 80-7169-860-1.
15. KUČERA, M. *HTML : tipy a triky od profesionálů*. Brno : UNIS Publishing, 2001. 80 s. ISBN 80-86097-64-1.
16. MEYER, E. A. *Eric Meyer o CSS : kompletní průvodce*. Brno : Zoner Press, 2007. 560 s. ISBN 978-80-86815-64-0.
17. MIKLE, P. *Dynamické HTML : referenční příručka*. Vyd 2. upr. Brno : UNIS Publishing, 2001. 168 s. ISBN 80-86097-09-9.
18. MIKLE, P. *Referenční příručka HTML*. 1. vyd. Brno : UNIS Publishing, 1996. 56 s.
19. MIKLE, P. *Rozšířené dynamické HTML : referenční příručka*. Vyd. 1. Brno : Zoner Press, 2004. 208 s. ISBN 80-86815-01-3.
20. MLÝNKOVÁ, I... [et al.]. *Technologie XML*. Praha : Karolinum, 2006. 186 s. ISBN 80-246-1272-0.
21. NAUMANN, F. *Dějiny informatiky : od abaku k internetu*. Vyd. 1. Praha : Academia, 2009. 424 s. ISBN 978-80-200-1730-7.

22. NEGROPONTE, N. *Digitální svět = Being digital*. Vyd. 1. Praha : Management Press, 2001. 208 s. ISBN 80-7261-046-5.
23. PALA, K. *Úvod do systému LaTeX*. 1. vyd. Praha : ČVUT, 1990. 66 s.
24. PÍSEK, S. *HTML : Tvorba jednoduchých internetových stránek*. 2. akt. a dopl. vyd. Praha : Grada, 2006. 108 s. ISBN 978-80-247-1767-8.
25. PÍSEK, S. *HTML & XHTML : začínáme programovat : podrobný průvodce začínajícího uživatele*. 1. vyd. Praha : Grada, 2003. 256 s. ISBN 80-247-0571-0.
26. RACHŮNEK, L. *TeX a Plain*. 1. vyd. Olomouc : Univerzita Palackého, 2003. 64 s. ISBN 80-244-0553-9.
27. RYBIČKA, J.; PŘIKRYL, P. *Programové vybavení počítačů : systémy pro počítačovou sazbu*. 1. vyd. Brno : Vysoká škola zemědělská, 1992. 98 s. ISBN 80-7157-057-5.
28. SKLENÁK, V... [et al.]. *Data, informace, znalosti a Internet*. 1. vyd. Praha : C. H. Beck, 2001. 510 s. ISBN 80-7179-409-0.
29. ŠPINAR, D. *Tvoříme přístupné webové stránky*. 1. vyd. Brno : Zoner Press, 2004. ISBN 80-86815-11-0.
30. ŠPINAR, D... [et al.]. *Přístupnost webových stránek orgánů státní správy*. Brno, 2007. 106 s. ISBN 978-80-903786-4-3.
31. VLASÁK, R. *Světové informační systémy a služby : informační průmysl*. 1. vyd. Praha : Karolinum, 1993. 178 s. ISBN 80-7066-801-6.
32. VLASÁK, R. *Světový informační průmysl*. 1. vyd. Praha : Karolinum, 1999. 342 s. ISBN 80-7184-840-9.
33. YOUNG, M. J. *XML krok za krokem*. Vyd. 1. Praha : Mobil Media, 2002. 472 s. ISBN 80-86593-28-2.
34. ZELENÝ, J.; MANNOVÁ, B. *Historie výpočetní techniky*. 1. vyd. Brno : Scientia, 2006. 184 s. ISBN 80-86960-04-8.
35. ZRZAVÝ, J. *VRML : tvorba dokonalých WWW stránek : podrobný průvodce*. Vyd. 1. Praha : Grada, 1999. 204 s. ISBN 80-7169-643-9.
36. ŽÁRA, J. *Jazyky pro popis virtuální reality*. Vyd. 1. Praha : ČVUT, 2000. 156 s. ISBN 80-01-02100-9.

Tištěné publikace v angličtině

37. ADOBE SYSTEMS. *Postscript language tutorial and cookbook*. 2nd Ed. Reading, Mass. : Addison-Wesley, 1985. 242 s. ISBN 0-201-10179-3.
38. ABRAHAMS, P. W.; BERRY, K.; HARGREAVES, K. A. *TeX for the impatient*. 1st Ed. Reading, Mass. : Addison-Wesley, 1990. 357 s. ISBN 0-201-51375-7.
39. BROPHY, P. *The library in the twenty-first century : new services for the information age*. 1st Ed. London : Library Association Publishing, 2001. XVI., 224 s. ISBN 1-85604-375-4.
40. BURKE, S. M. *RTF : pocket guide*. 1st Ed. Sebastopol : O'Reilly, 2003. 149 s. ISBN 0-596-00475-3.
41. GRAHAM, I. S. *The HTML sourcebook : a complete guide to HTML 3.0*. 2nd Ed. New York, NY : John Wiley & Sons, 1996. 688 s. ISBN 0-471-14242-5.
42. KIDDER, G., HARRIS S. *HTML publishing with Internet Assistant : Your guide to using Microsoft's HTML add-on*. 1st Ed. Chapel Hill, North Carolina : Ventana Press, 1995. 260 s. ISBN 1-56604-273-9.
43. KNOBLOCH, M.; KOPP, M. *Web design with XML*. [1st Ed.] Chichester : Wiley, 2003. 230 s. ISBN 0-470-84718-2.

44. KNUTH, D. E. *The TeXbook*. 9th rev. ed. Reading, Mass. : Addison-Wesley, 1990. ix, 483 s. ISBN 0-201-13448-9.
45. NIELSEN, J. *Hypertext and hypermedia*. 1st Ed. San Diego : Academic press, 1990. 268 s. ISBN 0-12-518410-7.
46. REID, B. K. *Scribe : introductory user's manual*. 1st Ed. Pittsburgh : Carnegie Mellon University, 1978. 76 s.
47. YADAV, A. *Deploying HTML5*. 1st Ed. Lexington, 2010. 224 s. ISBN 978-1-451-58954-2.

Elektronické zdroje v češtině

48. ANGELOVIČ, J. Popis SGML deklarace pro ISO/IEC15445. *Angelovic.cz* [online]. 2004– [cit. 2011-06-22]. Dostupný z WWW: <<http://www.angelovic.cz/internet/sgml-deklarace.html>>.
49. CRHONEK, V. *Historie značkovacích jazyků* [online]. Brno : Fakulta informatiky, Masarykova univerzita, 2003– [cit. 2011-06-22]. Dostupný z WWW: <<http://www.fi.muni.cz/usr/jkucera/pv109/2003/xcrhonek.htm>>.
50. Google [online]. Mountain View (CA) : Google, 1998– [cit. 2011-06-22]. Česká mutace. Dostupný z WWW: <<http://google.cz/>>.
51. HASSMAN, M. *HTML 4 5 6... : blog zabývající se vývojem HTML a XML* [online]. 2007– [cit. 2011-06-22]. Dostupný z WWW: <<http://html456.blogspot.com/>>.
52. CHAMURAPPI (pseudonym). *Webylon* [online]. 2004– [cit. 2011-06-22]. Dostupný z WWW: <<http://webylon.info/>>.
53. *Interval.cz* [online]. 1999– [cit. 2011-06-22]. Dostupný z WWW: <<http://interval.cz/>>.
54. *IT slovník : počítačový slovník* [online]. [cit. 2011-06-22]. Dostupný z WWW: <<http://it-slovník.cz/>>.
55. JANOVSÝ, D. *Jak psát web : o tvorbě, údržbě a zlepšování internetových stránek* [online]. 1998– [cit. 2011-06-22]. Dostupný z WWW: <<http://www.jakpsatweb.cz/>>. ISSN 1801-0458.
56. KAHOUN, P. Vývoj značkovacího jazyka HTML z hlediska sémantiky. *Inflow : information journal* [online]. 2007– [publ. 2008-02-10; cit. 2011-06-22]. Dostupný z WWW: <<http://www.inflow.cz/vyvoj-znacikovaciho-jazyka-html-z-hlediska-semantiky>>. ISSN 1802-9736.
57. KOSEK, J. *Domovská stránka Jirky Koska : vše o WWW* [online]. 1999– [cit. 2011-06-22]. Dostupný z WWW: <<http://www.kosek.cz/>>.
58. KOSEK, J. *HTML5* [online]. 1997– [cit. 2011-06-22]. Dostupný z WWW: <<http://htmlguru.cz/>>.
59. KOVÁŘ, P. *Historie výpočetní techniky v Československu : vývoj výpočetní techniky od roku 1950 : elektronkové a tranzistorové počítače, přehled generací počítačů* [online]. 2005– [cit. 2011-06-22]. Dostupný z WWW: <<http://historiepocitacu.cz/>>.
60. MACICH, J. ml. *Browsersy.cz : server o webových prohlížečích* [online]. 2009– [cit. 2011-06-14]. Dostupné z WWW: <<http://www.browsersy.cz/>>. ISSN 1803-814X.
61. OPÁLKOVÁ, M. *Markéta Opálková – osobní stránky : tvorba* [online]. 2009– [cit. 2011-06-14]. Dostupné z WWW: <<http://www.markytka.cz/tvorba>>.
62. PETERKA, J. *EArchiv* [online]. 2001– [cit. 2011-06-14]. Dostupné z WWW: <<http://www.earchiv.cz/>>.
63. ŠPINAR, D. *Pravidla přístupnosti* [online]. 2006– [cit. 2011-06-22]. Dostupné z WWW: <<http://www.pravidla-pristupnosti.cz/>>.
64. *Wikipedie : otevřená encyklopedie* [online]. St. Petersburg (Florida) : Wikimedia Foundation, 2001– [cit. 2011-06-22]. Česká verze. Dostupná z WWW: <<http://cs.wikipedia.org/>>.

65. Zdroják.cz : denní zpravodajství pro webové vývojáře, kodéry a webdesignery [online]. 1998– [cit. 2011-06-22]. Dostupný z WWW: <<http://zdrojak.root.cz/>>.

Elektronické zdroje v angličtině

66. BUSH, W. As we may think. *The Atlantic Monthly*. 1945, vol. 1945, is. 7. Dostupný také z WWW: <<http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/3881/>>.
67. CROCKFORD, D. Scribe. *No fluff just stuff* [online]. 2007– [cit. 2011-06-22]. Dostupný z WWW: <http://www.nofluffjuststuff.com/blog/douglas_crockford/2007/06/scribe>.
68. FIERZ, W. GRUETTER, R. The SGML standardization framework and the introduction of XML. *J Med Internet Res*. 2000, vol. 2, is. 2. Dostupný z WWW: <<http://www.jmir.org/2000/2/e12/>>. DOI 10.2196/jmir.2.2.e12.
69. GOLDFARB, CH. F. *The roots of SGML : a personal collection* [online]. 1996– [cit. 2011-06-22]. Dostupný z WWW: <<http://www.sgmlsource.com/history/roots.htm>>.
70. GRIFFIN, S. *Internet pioneers* [online]. [cit. 2011-06-22]. Dostupný z WWW: <<http://www.ibiblio.org/pioneers/index.html>>.
71. *HTML5 doctor helping you implement HTML5 today* [online]. 2011– [cit. 2011-06-22]. Dostupný z WWW: <<http://html5doctor.com/>>.
72. *IBM library server : GML starter set reference* [online]. 1989– [cit. 2011-06-22]. Dostupný z WWW: <<http://publibfp.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/dsm05m00/CONTENTS?SHELF=&DT=19910520154851>>.
73. LEPPÄNEN, M. *Differences between SGML, XML and HTML* [online]. Tampere : Hypermedia laboratory, Tampere University of Technology. 2000– [cit. 2011-06-22]. Dostupný z WWW: <<http://www.students.tut.fi/~leppane7/leppanen.html>>.
74. *Mark up poet* [online]. 2010– [cit. 2011-06-22]. Dostupný z WWW: <<http://www.markuppoet.com/>>.
75. NELSON, T. *Download XanaduSpace* [online]. 2007– [cit. 2011-06-22]. Dostupný z WWW: <<http://xanarama.net/>>.
76. PILGRIM, M. *Dive into HTML5* [online]. 2009– [cit. 2011-06-22]. Dostupný z WWW: <<http://diveintohtml5.org/>>.
77. SPRING, M. *Introduction to SGML* [online]. Pittsburgh (PA) : School of Information Sciences, University of Pittsburgh. 1997– [cit. 2011-06-22]. Dostupný z WWW: <<http://www.sis.pitt.edu/~spring/cas/node40.html>>.
78. TARABORELLI, D. *The beauty of LaTeX* [online]. [cit. 2011-06-22]. Dostupný z WWW: <<http://nitens.org/taraborelli/latex>>.
79. *W3schools online web tutorials* [online]. 1999– [cit. 2011-06-22]. Dostupný z WWW: <<http://www.w3schools.com/>>.
80. *Web Hypertext Application Technology Working Group : maintaining and envolving HTML since 2004* [online]. 2004– [cit. 2011-06-22]. Dostupný z WWW: <<http://www.whatwg.org/>>.
81. *World Wide Web Consortium* [online]. [1994–] [cit. 2011-06-22]. Dostupný z WWW: <<http://www.w3.org/>>.